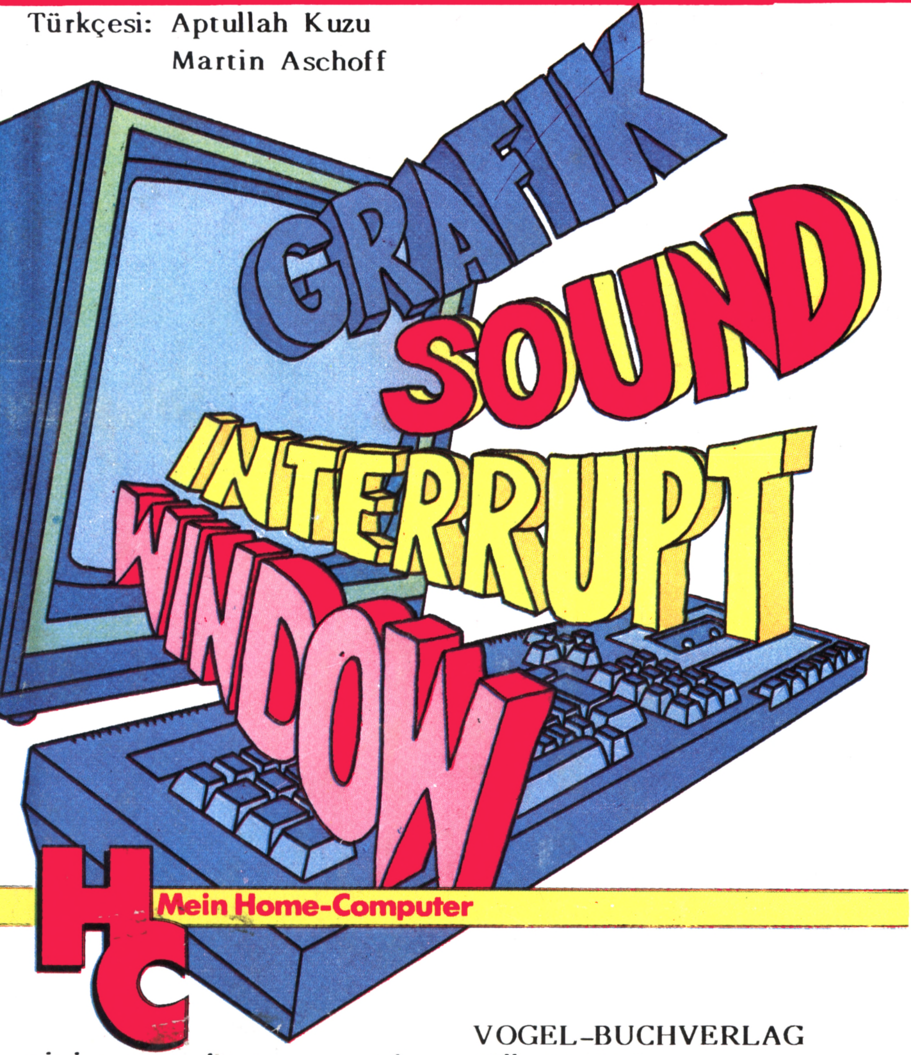
AMSTRAD CPC 464 NELER YAPABİLİR

Türkçesi: Aptullah Kuzu
Martin Aschoff



HC Mein Home-Computer

BİLİM TEKNİK YAYINEVİ

VOGEL-BUCHVERLAG
WÜRZBURG

Martin Aschoff

Amsrad CPC 464 Neler Yapabilir?

HC – Mein Home-Computer

Martin Aschoff

AMSTRAD CPC 464 NELER YAPABİLİR

Grafik, Ses, Kesilme, Ekran

Türkçesi: Aptullah Kuzu

BİLİM TEKNİK YAYINEVİ

VOGEL-BUCHVERLAG
WÜRZBURG

Was der CPC 464 alles kann-Martin Aschoff,
1985 baskısından çeviri yapılmıştır.

Baskı: DOĞAN OFSET

BİLİM TEKNİK YAYINEVİ
Bilgisayar Yayınları: 4

Cengiz Topel Cad. No: 13
Eskişehir, Tel. (9-221-22490)

Yazışma:
P.K. 75 Bahçelievler-İstanbul

İÇİNDEKİLER

Ö N S Ö Z

CPC 464 bilgisayarı, en çok satılan ev bilgisayarı olma yolundadır.

Ancak, tecrübeli kullanıcıya da yardımcı olabilecek, anlaşılır ve kapsamlı bir kullanım kitabının eksikliği hissedilmektedir.

Bu kitap, bilgisayarınızı daha iyi anlamanıza ve ondan daha çok yararlanmanıza yardımcı olacaktır.

Her CPC 464 bilgisayarı sahibinin pek çok yeni ve önemli bilgiler bulacağı bu kitap, CPC 464 kullanım broşüründen çok ileri kapsamlıdır.

Kullanım broşürünü tamamen incelemiş olanların kafasında dahi birçok soru cevapsız kalmış olacaktır. Bilgisayarları için ek bilgi, yeni fikir ve program arayışı içinde olanların bu arzularını karşılamak üzere, bu kitap üç büyük bölüme ayrılmıştır.

Birinci bölümde CPC 464'ün donanımı incelenmektedir. Bilgisayarın çalışma tarzına daha ayrıntılı bir bakış, sizi cihazınızla daha yakınlaştıracaktır. Bütün bağlantı noktalarının anlamları açıklanmakta ve çevre birimlerinin uzatma bağlantıları anlatılmaktadır.

Kitabın ikinci bölümü ayrıntılı olarak CPC 464'ün bellenimine ayrılmıştır. Burada BASIC'le programlamanın ve çalışma sisteminin inceliklerine değinilmiştir. Ayrıca CPC 464'ün diğer ev bilgisayarlarında olmayan bazı özelliklerinin nasıl değerlendirilebileceğine dair örnekler verilmiştir (Grafik, ses, kesilme ve pencereleme).

Kitabın üçüncü bölümü CPC 464 yazılımını sunmaktadır. Bir bilgisayarın değeri ancak yazılım zenginliğiyle ölçülebileceğinden, birçok faydalı ve gelişmiş programlar tanıtılmaktadır.

Kitap, hareketli grafik ve sesli oyunların programlanmasına dair ilginç bir kesimle son bulmaktadır.

Verilen programlar tümüyle denenmiş olup, bilgisayarın orijinal çıkışlarıdır. Bu nedenle hepsi CPC 464'le kusursuz çalışacaklardır.

Darmstadt — Martin Aschoff

G İ R İ Ş

CPC 464'le bilgisayar pazarı, kendisinden beklenileni veren bir ev bilgisayarına kavuşmuştur. Bu bilgisayar satın alındığı andan itibaren kullanıma hazırdır, çünkü teybi ve denetleyicisi satış fiyatının içindedir. Buna rağmen emsallerinden daha pahalı değildir.

CPC 464 herkes için uygun bir bilgisayardır. Oyunları sevenler için renkli denetleyici olduğu gibi, iş adamları için de 80 işaretlik satırlarıyla tek renkli denetleyicisi, yazdırıcısı ve CP/M çalışma sistemli disket kutusu vardır.

Geniş bir satış ve servis ağının yanısıra, fiyatına göre yüksek hizmet kapasitesi ve imalatındaki kaliteli işçiliği de CPC 464'ün sevilmesini sağlamıştır. Ancak CPC 464'ün hızla yayılmasının asıl nedeni, onu rakiplerinden ayıran birçok dikkat çekici özelliğinin olmasıdır. Bunların başlıcaları şunlardır:

- 128 000 resim noktasıyla yüksek çözümleyici grafik
- 27 rengi verme imkânı
- 8'e kadar ekran penceresine imkân veren pencere teknolojisi
- Birbirinden bağımsız üç kanallı ses üreteci, 5-ses-bekleme sırası ve gürültü üreteci,

□ BASIC dilinde çok iyi kesilme imkânları.

İmalâtçı firmanın CPC 464'ün bu özelliklerine yeterince değinemediği anlaşılmaktadır. Bilgisayarla birlikte verilen kullanma talimatnamesi, 280 sayfa uzunluğunda olmasına rağmen, sık sık tekrarlar ve bazı noktalarda da yanıltıcı bilgiler içermektedir.

Kullanma broşürünün önsözünde, CPC 464'ün ayrıntılarına dair bilgilerin kitaplar dolusu olacağına işaret edilmektedir. Bu biraz abartılmış olabilir. Ancak imalâtçının gözden kaçırdığı noktaları belirtmek ve kullanıcıyı aletin bütün imkânlarıyla tanıştırmak gerekirse, rahatlıkla bir kitabı dolducak bilgi ortaya çıkmaktadır.

İşte bu kitap şimdi elinizdedir! Bu kitabın dikkatli bir şekilde okunması, size CPC 464'ün kullanımında yeni ufuklar açacaktır.

1

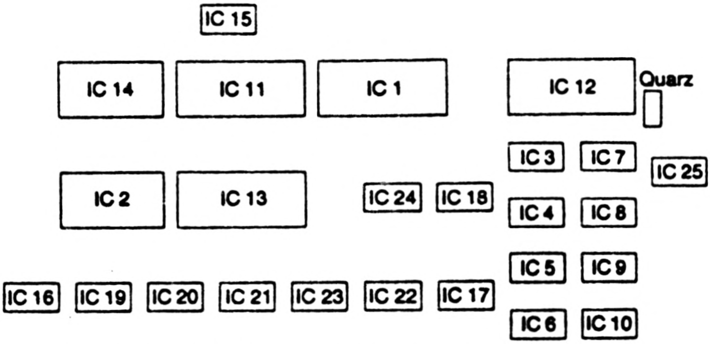
CPC 464'ün DONANIMI

Bir bilgisayarın donanımı, mekanik ve elektronik parçalarından oluşur. Deyim yerindeyse, dokunabileceğiniz her şey donanıma aittir.

Önce CPC 464'ün içerden nasıl göründüğünü anlatıp, sonra önemli parçalarına dair bilgi vereceğiz.

1.1. Bilgisayarın İçi

Siz herhalde elektronik hastası değilsinizdir ve bilgisayarınızı parçalarına ayırmamışsınızdır. Bu nedenle bu bölümde CPC 464'ün içinin nasıl olduğunu ve orada ne olup bittiğini anlatacağız. Cihazın açılması nispeten problemsizdir. (Ancak cihazı açtığınız takdirde garanti haklarınızı kaybedip kaybetmedinize dikkat ediniz!). Bilgisayarın altındaki altı vidayı sökmek ve üst tarafını kaldırmak yeter. Fakat üst taraf, klavye ile hesaplama kiti arasındaki bağlantıyı sağlayan 20'lik bir yassı kablo ile hala bilgisayara bağlıdır. Üst tarafı tamamen kaldırmak için bu kabloyu kitden sökmeniz gerekir. Ancak kırmamak için sökerken zorlamayınız. Bilgisayarın üst kısmını kenara koyduğunuzda, artık CPC 464'ün kiti karşınızdadır. Onu muhafazanın alt tarafına bağlayan dokuz vidayı sökmenize gerek yoktur. 2 cm. kenarlı ve hemen dikkati çeken siyah bir küp, klavye yayı olarak hizmet görmektedir. Kit üzerinde birçok direnç, birkaç kondansatör ve bir miktar entegre devre görülmekte olup, entegre devrelerden biri soğutucu cisim altındadır.



Resim 1.1 Baskılı devre üzerine şematik bakış

IC-No.	Markası	Görevi
IC 1	Z-80A-CPU	4MHz darbe frekansı için mikro işlemci
IC 2	23256	32 KBayt-ROM
IC 3-10	4164	64-Kx1-Bit-dinamik RAM
IC 11	8255PIO	Paralel giriş/çıkış elemanı
IC 12	20RA043	Gate Array -Bellek yönetimi -RAM belleğin çalıştırılması -Video çıkış sinyali -Darbenin bölünmesi
IC 13	6845CRTC	Video sinyal jenaratörü (CRTC)
IC 14	AY-3-8912	Ton jenaratörü
IC 15	74LS273	Centronics-Kesim konumu
IC 16	74LS145	Kod çözücü
IC 17	74LS244	8-Bit veri okuma ünitesi
IC 18	74LS373	8-Bit veri yazma ünitesi
IC 19-22	74LS153	Video bellek CRTC için veri selektörü
IC-23	74LS132	Lojik Bağlantılar için VEDEĞİL kapısı
IC 24	74LS32	Lojik Bağlantılar için VEYA kapısı
IC 25	7400E	Darbe yönlerinin üretimi için VEDEĞİL kapısı
Quarz	HC-18/U	16 MHz darbe üretici kristal

Tablo 1.1 CPC 464 'de IC'lerin anlamı

Çok dikkatli bakacak olursanız üç transistör-bir diyot ve CPC 464'ün kuvarsını göreceksiniz. Bu devre elemanlarından bizi ilgilendirenler, sadece entegre devreler ve kuvarstır. Şekil 1.1'de bu parçaların dikkate alındığı bir kit şeması görülmektedir (IC 12, 12 numaralı entegre devre, çok fazla ısındığı için soğutucu bir metal altında bulunmaktadır). Entegre devrelerin ve kuvarsın anlamları Tablo 1.1'de açıklanmıştır. Kitin en sağında bulunan 8 kafalı bir fiş teyple olan bağlantıyı sağlamaktadır. Teybin elektroniği bizi fazla ilgilendirmemektedir. Giriş ve çıkış sinyalleri için iki yükseltece ilâveten bir de mikrofon yükselteci mevcuttur.

1.2 Mikro İşleyici

Modern bilgisayarların kalbi artık tek bir elemandan oluşmaktadır. Buna mikro işleyici veya CPU (merkezi işlem birimi) adı verilir.

CPC 464'ün mikro işleyicisi, Zilog firmasının mikro bilgisayarlar için çok yaygın olarak kullanılan bir ana işlem birimi olan Z-80A-CPU'dur. (Buradaki A, en yüksek çalışma frekansı olan 4 MHz için imalatçı firmanın kod harfidir). Mikro işleyici CPC 464'ün beynidir. O, verileri işler, aktarır, lojik ve aritmetik emirleri yerine getirir ve çevre birimleri üzerinden çevresiyle iletişim kurar.

Komutlar

Bir mikro işleyicinin anlayıp, yerine getirebileceği komutların sayısı, onun kapasitesi için önemli bir göstergedir.

Z-80-CPU mikro işleyicisi (A harfini, bizim için önemli olmadığından artık yazmıyoruz) birbirinden farklı 158 komuta cevap verir. (İşlem Kodları).

Yazmaç

Bir mikro işleyicinin yazmaçlarının sayısı da önemlidir. Mikro işleyicinin ne kadar çok yazmacı varsa, o kadar çok veriyi depolayabileceğinden, sonradan veri yüklemesine o kadar az ihtiyaç duyulur. İşleyicinin çalışma şekli de daha berrak, effektif ve hızlı olur. Z-80-CPU'nun 22 yazmacı vardır ki, bu yüksek bir sayıdır.

Veri Taşıma Kanalı

Mikro işleyicinin çevresiyle bilgi alışverişinde bulunabilmesi için, veri taşınacak bir kanala ihtiyacı vardır. Buna veri taşıma kanalı denir. Bu kanalın genişliği, mikro işleyicinin kapasitesi için bir göstergedir. Çünkü 16 Bitlik ve kanalda belli bir zamanda 8 Bitlik bir kanala göre iki misli daha çok veri taşınabilir.

Ev bilgisayarlarının çoğunun mikro işleyicilerinin veri kanalı 8 Bitliktir. Z-80-CPU da 8 Bitliktir. Daha büyük bilgisayarlarda 16 veya 32 Bitlik veri kanalları vardır.

İç Bilgi-İşlem Kapasitesi

Bir mikro işleyicinin veri kanalı kapasitesi kadar iç bilgi-işlem kapasitesi de önemlidir.

Z-80-CPU 8 Biti paralel işleyebilir ve bu ne-

denle 8-Bit işleyicileri ailesine dahildir. Bundan başka, uygun komutlarla 8 Bitlik iki yazmacı 16 Bitlik bir yazmaca dönüştürmek mümkündür. Bu nedenle Z-80-CPU 8-Bitlik gerçek bir işleyici olmanın yanısıra, 16 Bitlik bir varsayılan işleyicidir. Çünkü tek bir komutla 16 Bit uzunluğunda veri kelimeleri işlenebilir. Bu metodun faydası ortadadır:

İşlem kodları her defasında sadece bir tane yazmacı hedef alırlar. 8 Bit yerine 16 Bitlik yazmaçlar kullanılırlarsa, aynı bir zamanda iki misli fazla veri işlenebilir.

Bazı mikro işleyiciler, iç kapasiteleri veri kanalı genişliğinden daha büyük olduğundan, dışarıya bir defada verebileceklerinden daha çok veriyi paralel olarak işleyebilirler.

Örneğin yazmaçları 32 Bitlik, veri kanalı 8 Bitlik bir işleyicide dış donanım tasarrufu sağlanmış olur. Ancak bu şüphesiz işlem hızını yavaşlatır.

Adres Kanalı

Mikro işleyicinin bir veri taşıma kanalı yanında, çevresine mesaj gönderebilmek için bir de adres kanalına ihtiyacı vardır. Bu kanalın genişliği, aynı anda kaç tane çevre birimiyle görüşebileceğini belirler. Mikro işleyicilerin adres kanalı genişliği genellikle 16 Bittir ve böylece aynı anda 2^{16} yani 65536 çevre birimi idare edilebilir. Çevre birimleri tamamen bellek birimlerinden oluşuyorlarsa, mikro işleyici 65536 Byte, yani 64 Byte (1 KByte=1024 Byte) kullanabilir. Çok bilinen maksimal bellek kapasitesi olan 64 KByte böyle ortaya çıkar.

I/O-kapıları ile konuşulması genellikle oldukça komplikedir. Örneğin çok kullanılan 6502-CPU'da her kapı için bir bellek birimi rezerve edilir. Ancak Z-80-CPU'da durum böyle değildir. Burada tek bir komutla 256 kapıyla aynı zamanda görüşülebilir. Bu kapı adresi aşağı 8 adres biti üzerinden ekstra bir kapı sinyaliyle verilir. Böylece Z-80-CPU'da hem 64 KByte'lık tam bellek kapasitesi kullanılmış, hem de yeterince I/O-kapılarına yer ayrılmış olur.

Bu kapılar üzerinden uygun donanımla farklı bellek bloklarına geçilebilir ve neticede maksimal 256×64 KByte'lık bir kapasite ortaya çıkar.

CPC 464'ün Z-80-CPU'su da geçit-dizisi ile birlikte bu metodu kullanarak iç ve dış ROM ve RAM'ları ayırdedebilir.

Yönetme Kanalı

Mikro işleyicinin üçüncü kanalı olan yönetme kanalı, mikro işleyicinin durumunu gösterir. Bu kanalın sinyalleri belli bir anda okuma veya yazılma mı yapıldığını bellek veya I/O-kapılarıyla mı görüşüldüğünü veya çevreden bir kesilme talebi mi olduğunu vs. gösterirler.

Sinyal	Kayıt/verme durumu	Aktif durumdaki anlamı
D0	Kayıt/verme durumu	Veri kanalının en düşük değerli Bit'i yerleştirilir.
D1	Kayıt/verme durumu	Veri kanalının düşük değerli Bit'i yerleştirilir.
D2	" " "	" " " " " " "
D3	" " "	" " " " " " "
D4	" " "	" " " " " " " "
D5	" " "	" " " " " " "
D6	" " "	" " " " " " "
D7	" " "	Veri kanalının en büyük değerli Bit'i yerleştirilir.
A0	Verme durumu	Adres kanalının en düşük değerli Bit'i yerleştirilir.
A1	" "	Adres kanalının düşük değerli Bit'i yerleştirilir.
A2	" "	" " " " " "
A3	" "	" " " " " "
A4	" "	" " " " " "
A5	" "	" " " " " "
A6	" "	" " " " " "
A7	" "	" " " " " "
A8	" "	" " " " " " "
A9	" "	" " " " " "
A10	" "	" " " " "
A11	" "	" " " " " "
A12	" "	" " " " " " "
A13	" "	" " " " " "
A14	' '	' ' ' ' ' "
A15	" "	Adres kanalının en yüksek değerli Bit'i yerleştirilir
RD	Verme durumu	Veriler okunur.
WR	" "	Veriler yazılır
MREQ	" "	Adres kanalı belleğe uyar
IORQ	" "	Adres kanalının alt Bayt'ı I/O Port'a uyar.
RESET	Kayıt "	CPU'nun ön yüklenmesi
NMI	" "	Cereyanı kesilmeyen kesilme talebi
INT	" "	Cereyanı kesilen kesilme talebi
HALT	Verme "	CPU bir sonraki kesilmeyi bekliyor
WAIT	Kayıt "	CPU sinyal aktifliğini yitirinceye kadar bekler
BUSRQ	" "	CPU'nun cereyanı kesilmez, harici elemanlar
BUSAK	Verme "	Bellek ve I/O Port'u kavrarlar
BUSAK	Verme "	CPU cereyanın kesilmediğini tasdik eder.
RESH	" "	Adres kanalının alt yedi Bit'i dinamik RAM'ın Refresh adreslerini kapsar
M1	" "	İlk makina çalışır, yani CPU bellekten komut kodları alır.
	Kayıt "	Darbe girişi
+5V	" "	Besleme gerilimi girişi
GND	" "	Şase girişi

Tablo 1.2 Z-80-CPU'nun sinyalleri

Z-80-CPU'nun sinyallerini tablo 1.2'de görebilirsiniz. Sinyal üzerindeki bir çizgi, bu sinyalin negatif mantıkla çalıştığına işaret etmektedir. Yani sinyal aktif durumda 0 ve pasif durumda 1 üzerindedir.

Veri, adres ve yönetme kanalları için bağlantıların yanısıra, mikro işleyicinin besleme gerilimi için iki (+5V ve GND) ve ölçü için de bir girişi vardır.

Ölçü

Mikro işleyici bütün görevlerini aynı anda rürütemez, emirleri adım adım yerine getirmek durumundadır. Emirlerin icra zamanı bir ölçü frekansı ile belirlenir.

Mikro işleyicinin bir komutu işlediği zamana, bir komut devri denir. Bir komut devri, bir veya birkaç makina devrinden oluşur. Komut nekadar komplike ise, icrası için o kadar çok makina ~~makina~~ devri gerekir. İlk makina devrinde komut bellekten okunur ve kod çözülür, bundan sonraki devirlerde icra edilir.

Bir makina devri de birkaç ölçü devrinden oluşur ve bunların süresi ölçü frekansı ile belirlenir. Bir ölçü devri bir ölçü peryodu uzunluğundadır.

Bir komut devrinin ilk makina devri, şu ölçü devirlerine ayrılabilir: komutun CPU'ya yüklenmesi, program sayıcının bir yükseltilmesi, komut kodunun çözülmesi (ve basit komutların icrası) ve (bazı mikro işleyicilerde) dinamik RAM'lar için yeni ölçü verilmesi. Komut basit ise, komut devri sadece bu ilk makina devrinden oluşur. Aksi takdirde bunu diğer makina devirleri takip ederler.

Bir komut, komplikelik derecesine göre, birkaç ölçü devrinden oluşabileceği gibi, yüzlerce ölçü devri uzunluğunda da olabilir (meselâ 32 Bitlik virgüllü sayıların işaretli bölmesinde). Z-80-CPU mikro işleyicisinin en uzun komutları 23 ölçü devrinden oluşur. CPC 464'ün sistem ölçüsü olan 4 MHz'lik bir ölçü frekansında, 23 ölçü devri 5,75 µs'lik bir zaman alır.

1.3 Bellek

Bir bilgisayar için bellek, mikro işleyici kadar önemlidir. Bellek olmadığı takdirde, mikro işleyici sadece yazmaçlara sığacak kadar veriyi işleyebilir. Bellek bulunduğu takdirde, işlenebilecek veriler ancak bellek kapasitesiyle sınırlanır.

İnsanlardaki uzun, kısa ve çok kısalık hafızalar gibi, bilgisayarın da yapıları gereği farklı süreli depolamalar için uygun farklı bellek tipleri vardır.

Yazmaç

Mikro işleyicinin yazmacı insandaki çok kısa sürelik hafıza ile karşılaştırabilir. Buraya sadece direkt olarak gerekli olan bilgiler kaydedilir. Meselâ; bundan sonraki mantıksal ve aritmetik bağlantı değerleri, alt program yordamlarının adresleri, kaydırılması gereken veri bloğunun çalışma adresleri vs.

RASTGELE ERİŞİMLİ (ÇALIŞMA) BELLEK (RAM). Bilgisayarın kısaca RAM olarak adlandırılan çalışma belleği insanın kısa süreli hafızası gibidir. Bu bellekte bilgileri gerilim olduğu sürece bulunan yazma ve okuma bellekleri vardır.

RAM statik ve dinamik olarak sınıflandırılabilir. Statik bellek satırları flipflop gibi yapılmıştır. Değerler okunur ve çalışma gerilimi kesilinceye kadar kayıtta kalır.

Dinamik bellek satırı, prensip olarak sadece bir FET transistor (Alan etkili transistor) ve bir kondansatörden oluşmaktadır. Kondansatör yüklendiği zaman bellek satırının değeri 1, boşaldığı zaman 0'dır. Her yapı elemanı için statik RAM'dakinden daha yüksek yoğunlukta Entegreler bulunması, dinamik RAM'ın bir üstünlüğüdür.

Böylece CPC 464'ün dinamik RAM belleği sadece 8 entegreden oluşmuştur. Bu entegrelerin her birinde 64KB x 1 Bit'lik veriler organize edilebilir, yani her Bit için 65536 adres bulunmaktadır. Bu sekiz Entegre paralel bağlanırsa 64KBx8 Bit yani 64 KByte RAM elde edilir. Bu aynı zamanda CPC 464 çalışma belleğinin kapasitesidir. Bellek alanının üst kısımdaki 16 KByte'lik kısmı video ekranını içeren ekran belleği tarafından doldurulduğu için çalışma sistemi ve kullanıcı için sadece 48 KByte'lik alan bulunmaktadır. Dinamik RAM'ın tek mahsuru dinamik RAM'a nazaran daha karmaşık kullanılmasıdır. Dinamik bellek satırının kondansatörü zamanla akımın akması ile yükünü kaybedeceğinden yaklaşık 2 milisaniyede yeniden yüklenmesi gerekir. Bu durum Refresh (canlandırma) olarak tanımlanır.

Modern mikro işlemciler bilgi ve adres kanalını kullandıkları zaman dinamik RAM'ın Refresh'ini korurlar. Bu amaç için Z-80-CPU da özel bir Refresh devresi vardır. Bu devrenin değeri periyodik olarak bir yükseltilir ve adres kanalının alt 7 Bit'lik alan üzerine Refresh adresleri verilir. Bir bilgisayarın RAM belleği şu anda çalışan işlemekte bulunan programı, alfa nümerik bilgileri ve çalışma sisteminin özel bilgilerini içerir.

Sabit Değer Belleği (ROM)

Devamlı kullanılan ve unutulmaması gereken bilgiler bilgisayarın sabit değer belleğinde bulunmalıdırlar. Bunlar ROM, PROM, EPROM ve EEPROM olarak ayrılırlar. Bütün bu bellek tiplerinin ortak bir özelliği vardır: Çalışma gerilimi kesildiği anda silinmezler aksine içeriklerini korurlar.

ROM (salt okunur bellek), içeriği üretimi sırasında tesbit edilen bir sabit değer belleğidir. Her bir bellek satırı iki belirli noktanın iletken yada iletken olmayan bağıntısına uyar.

PROM (Programlanabilir salt okunur bellek) içeriği müşterisi tarafından tesbit edilebilen bir sabit değer belleğidir.

EPROM (silinebilen programlanabilir salt okunur bellek) aynen PROM gibi bazı bellek satırlarına yüksek gerilimlerin bağlanmasıyla programlanabilir. PROM'un aksine EPROM silinebilmektedir. Şayet entegrenin yüzeyleri yaklaşık yarım saat yada bir saat bir UV-lamba ile aydınlatılırsa, içindeki bilgileri kaybeder. Bu da onun silinebilirlik özelliğidir.

EEPROM'un (elektriksel silinebilen programlanabilir salt okunur bellek) EPROM'dan ayrı olan bir üstünlüğü ise, elektriki olarak silinebilmesi ve kullanıcının hangi bellek satırlarının silinmesi gerektiğini tespit edebilmesidir (EPROM ise EEPROM'un aksine tamamen silinmektedir). Bunun için EEPROM en pahalı sabit değer belleğidir.

Harici Bellekler

Buraya kadar anlatılan bellek tipleri direkt olarak bilgisayar muhafazasının içersinde bulunur ve dahili bellekler olarak adlandırılır. Bilgisayarın en büyük bellekleri ise cihaz haricinde bulunur ve harici bellekler olarak tanımlanır. Buna kitaptan diskete kadar veri taşıyıcıların tüm şekilleri dahildir.

Burada kitabı da belirttiğimize belki şaşıracaksınız ama, kitap ta bir bellektir, zira bazı bilgiler içermektedir. Bilgisayarınız sadece kitapla oluşamaz. Önce bilgileri değiştirmeli ve bilgisayara okunacak şekilde sunulmalıdır. Meselâ bunu kitaptaki bir program listesini CPC 464'e yüklemekle başarabilirsiniz. Böylece kaset ve disketlerin manyetik veri taşıma özelliği bellekte sınırlandırılmış olur.

Kaset zamanımızda en kolay genişletilebilen bir bellek aracıdır. Bir kaset yaklaşık 1 DM olmasına rağmen, üzerine 1 MBayt'a kadar bilgi alabilmektedir. Ayrıca kaset teyp'in de fiyatı oldukça ucuzdur. Bu bakımdan kaset-bellekler oldukça ilgi çekmektedir.

Kasetin mahsurları ise işaret yoğunluğunun azlığı ve taşıma hızının düşük olmasıdır. İlk bilgisayarlar 300 Baud (Bit/saniye) ile çalışırken, standart bilgisayarlar yaklaşık 1200 Baud'a ulaşmaktata, disketlerin hızı ise su hızını geçmemektedir. Ayrıca kasetin arzu edilen programın başladığı yere kadar uzunca bir süre sarılması gerekir. CPC 464 2000 Baud hıza ulaşabilmekte ve ayrıca hızlı sorabilen motoru ile övülmeye değer bir özellik arzetmektedir.

Disket kasetten % 10 daha hızlıdır. Bu bellek aracı kasete nazaran dakikada birkaç yüz devir daha fazla dönmektedir.

Ayrıca bir disket kasetin aksine daha kesin şekillidir. Disket yine hususi sektörlerden (alanlardan) oluşan daha fazla bölüme ayrılmıştır. Bir sektör 128 Bayt (tek yoğunluk) yada 256 Bayt (çift yoğunluk) içerir.

CPC 464'ün disketi 180 KBayt'a kadar bilgileri taşıyabilir. Disketin herbiri 128 Bayt'lık 36 sektörlü 40 bölümü vardır, böylece basit yazı yoğunluğu ile çalışır (tek yoğunluk ile çalışır).

Disket kullanmanın mahsuru disketin yüksek fiatıdır. Sadece tek yüzünden yararlanılabilinen tek ses başlıklı bir disket yaklaşık bir bilgisayar fiatı kadardır. Çift yüzeyi kullanılabilen ve çift ses başlıklı bir disket tabii ki çok daha pahalıdır.

CPC 464'ün disketi ana cihaz kadar pahalıdır. Bunun için daha rahat genişletilebilen CP/M çalışma sistemi ve Logo programlama dili bulunan sistemleri tercih ediniz.

1.4 Kaset-Teyp

Programların ve bilgilerin kaydedilmesine yarayan kaset-teybin cihaz üzerinde bulunması CPC 464'ün bir üstünlüğüdür. Böylece komputerin ve elektronik teybin birbirine optimal bir şekilde uyması sağlanmış herhangi bir adaptasyon problemleri oluşmasını engellemiştir. Kaset başa yada sona doğru sarılırken keskin bir ses çıkarır ve EJECT tuşuna ani bir şekilde basarsak, teybe yerleştirdiğimiz kaset yukarı doğru fırlar. Bu durumda yayın biraz gevşetilmesi gerekir. Bu özelliklerden de görüldüğü kadarıyla teybin mekaniği sağlam ve çalışması düzenlidir. Elektronik kısımda oldukça güvenilir olmasına rağ-

men düşük değerli kasetlerde okuma hataları oluşabilir.

Baud - Hız Bölümleri

Programlayıcı, üzerinde iki farklı bilgi aktarma hızı bulunmaktadır. Birincisi 1000-Baud'luk ikincisi ise daha hızlı 2000 Baund'luktur.

Hesaplama esasında 8,2 yani her KBit için 4,1 saniyelik süre okuşur. Bu değere pratikte ulaşılamaz. Zira CPC 464'ün verileri 2 KBit'lik sabit bir formda kaydedilmektedir. 1000 Baud'luk çalışma şeklindeki formda ön gerilim, 7 saniye tanıma, 17 saniyede bilgiler için gereklidir.

2000 Baud'luk çalışma şeklinde de 3 saniye ön gerilim, 3,5 saniye tanıma, 8,5 saniye de bilgiler için gereklidir.

Bu çalışma şekli 2 KBit'lik blok'un tam bir şekilde belirtilme hususunda geçerlidir.

1000 Baud'luk bilgi taşıma bölümünde CPC 464'ün 2 KBit için 27 saniyeye 2000 Baud'luk hızda ise komple blok için toplam 15 saniyeye ihtiyaç vardır. Gerçekte sadece 17 saniye yada 8,5 saniye süresince bilgiler aktarıldığı için CPC 464'ün aktarma bölümü oldukça yavaş oluşacaktır. 1000 Baud'luk bölümde zamanın % 63'ü 2000 Baud'luk bölümde % 57'si bilgi aktarmaya harcanır.

Sizin için hangi bölüm daha iyidir? Bilgiler ve programların bir kopyasını yapan programlayıcı 2000 Baud hızlık bölümde güvenilir bir şekilde çalışabilir. Bu kopye şayet istisnai bir şekilde bir okuma hatası oluşursa kullanabilir.

Zamandan dolayı ilave bir kopya hazırlamaktan kaçınılırsa 1000 Baud'luk daha yavaş ve emin olan bölüm seçilmeli ve en iyi. en kaliteli kaset kullanılmalıdır.
(Dikkat: Kromdioksit kasetler kullanmayınız!)

1.5 Harici Donanım

Harici donanım olarak direkt ana cihaza bağlı olmayan ve mikro işleyici bellek, giriş-çıkış ports'-undan oluşan bütün birimleri belirtilmektedir. CPC 464'ün harici donanımı olarak tuş takımı, kaset-teyp ve monitordur.

Tuş Takımı

CPC 464'ün tuş takımı zedelenmeyen ve rahat kullanılan 74 özel tuştan oluşmaktadır. Bu tuş takımı yapımı itibariyle oldukça tesirli ve pratikteki kullanımı da daha kolaydır.

CPC 464'ün tuş takımı Z ile Y harfinin yer değiştiği ve Alman umlautlarının olmadığı bir Amerikan stili bir tuş takımıdır.

KEY-DEF komutu ile Z ve Y harflerinin yeri değiştirilebilir, SYMBOL komutu ile alman umlautları belirtilmelidir. (Bakınız: Konu 2.4.4)

Kaset-Teyp

Kaset-Teyp son bölümde geniş bir şekilde anlatılacaktır.

Monitor

CPC 464'ün en enteresan harici donanım birimi

şüphesiz ki monitordür. Cihaz satın alınırken tek renkli yeşil monitor yada renkli monitor seçilebilinir.

Her monitorün avantaj ve dezavantajları vardır.

Her iki monitorde CPC 464'ün şebeke enerjisini sağlarlar. Dikkat: Monitor olmadığı zaman komputerde akım yoktur.

Monitorün şebeke enerji kısmı her iki durumda da 3 Amperlik bir çıkışı akımı üretir. Bu 5 Volt'luk gerelim iletiminde 15 Watt'lık çıkış gücüne uygundur. CPC 464 için sadece 10 Wattlık bir güce gereksinim vardır. Bunun da 1 Watt'lık kısmı kaset-teybe harcanır. Böylece özel amaçlar içinde 5 Watt'lık bir yedek güç bulunmakta, yani harici devreler için 7 Ampere kadar akım kullanabilirsiniz. Böylece CPC 464 gerilim sağlama işlemini derhal kesmeyen birkaç komputerden biridir. Cihazın genişleme (yayılma) kısmındaki iki adet Led-Diyot çalışmaktadır. (Bakınız: Konu 1.6.1)

Modülatör

Aynı şekilde kesici kısımdan bir Hf-Modülatör elde edilebilir. Modulatör CPC 464'ün Monitör çıkış sinyalini televizyon cihazının tanıyamıyacağı (algılayamayacağı) şekilde değiştirebilir. Sadece yeşil monitörünüz varsa bu durumda cihaza bir renkli televizyon cihazı bağlayabilirsiniz.

Modülatörde şebeke kısmının ne aradığını kendi kendinize belki düşüneceksiniz. Gördüğümüz gibi CPC 464 monitörüne bağlıdır. Şayet bir monitör bağlamazsak, modülatör akımı temin etmeyecektir. Televizyon cihazının bağlantısı için neden bir modülatöre ihtiyaç vardır? Normal bir anten kablosu yeterli değil midir?

Basitçe tarif edilecek olursa modülatör bir yönetim birimi ve bir ekran tüpünden oluşmaktadır.

Yönetim birimi giriş sinyalinden video resmi teşekkül eder ve ekran tüpü üretilen bu resmin görünmesini sağlar. Renkli cihazda değişik gönderileri algılayabilen bir Tuner (alıcı) bulunur. Bu Tuner her antenin gönderdiği dalga karışımlarından belirli bir frekansı seçer ve taşıyıcı dalganın Demodülasyonu ile, yönetim birimi tarafından oluşturulan video resmi üretir. Televizyon cihazı aslında alıcılı bir monitörden başka bir şey değildir.

Televizyon cihazının Tuner'inin mahsuru monitör için belirli video sinyali direkt bilgisayardan almaması, aksine taşıyıcı dalgayı üreten ve video sinyalini bu dalgaya modüle eden bir modülatöre gereksinim duymasıdır. Televizyon cihazının Tuner'i modülasyon aracılığıyla taşıyıcı dalgayı ve yararlı sinyali ayırdeder. Bu teknik yol ve televizyonun modülatör karşısındaki zayıflığının (18 MHz'e karşılık 5,5 MHz) sebebi; renkli resmin 50 işaret/satırdan itibaren net olmaması ve karışmasıdır. Ciddi kullanımlarda (meselâ metin işleme programında) bundan dolayı monitör gereklidir.

Basıcı

Ekranın yanında bilgisayarın en önemli kayıt/verme birimi basıcıdır. Genelde iki tip basıcı vardır. Matris basıcı ve matbaa tipi basıcı. Matris basıcılar basılması gereken her işareti nokta matrislerden oluştururlar. Bir nokta matrisinin nokta sayısı bu durumda yazı resminin niteliğine bağlıdır.

Matbaa tipi basıcılar tarif edilen bütün işaretler için daire şeklindeki baskı damgaları kullanırlar. Matbaa tipi basıcı böylece tuş takımı olmayan elektrikli bir daktiloya benzemektedir.

Bilindiği gibi her sistemin yararları ve mahsurları vardır. Matris basıcıların üstünlüğü yüksek baskı hızı, düşük ücret ve noktalardan oluşan grafiklerin ifade edilebilme kabiliyetidir. Buna karşın matbaa tipi basıcılar üstünlüğü oldukça temiz yazı resmi ve yazı tarzının çabuk bir şekilde değiştirilebilmesidir.

Gelecekte Matris basıcılar daha çok kullanılacaktır, zira matbaa tipi basıcıdan 50 kez daha hızlı, fiatı ise yarı yarıya ucuzdur.

Ayrıca modern Matris basıcılar kabul edilen bir yazı şeklinin hakkını vermeye çalışır. Baskı kalitesi için nokta matrisin inceltilmesi ve baskı hızının düşürülmesiyle, matbaa tipi basıcıdan çok zor ayırdedilebilen bir yazının oluşturulması mümkündür. Bu baskı kalitesi genelde bitişik harf kalitesi (NLQ) olarak belirtilir. En yeni matris basıcılar matbaa tipi basıcılardan ayırdedilmeyecek bir yazı resmi verebilmektedirler. Bunlar zamanımızda birkaç bin mark daha pahalıdır ve kalitelerini (LQ) ile belirtirler.

Schneider firması tarafından CPC 464 için sunulan NLQ 401 basıcısı isminden de anlaşıldığı gibi bir NLQ moduna sahiptir. Baskı hızı en hızlı durumda saniyede 50 işarettir. Çok fazla bilgiler basmak yada profesyonel metinleri işlemek istiyorsanız daha konforlu bir basıcı ile çalışabilirsiniz.

Planlayıcı

Matris basıcının grafik imkânları size yeterli gelmiyorsa, bir plânlayıcı kullanmanız gerekecektir. Bu cihaz iki yöne hareketli ucuyla tabii işaretleri yazabilir. Bunların tek renkli, çok renkli ve 1 mm'den 0,05 mm kadar incelebilen yazı karakterli tipleri vardır.

Diğer Çevre Elemanları

Bu arada CPC 464 için çeşitli harici çevre elemanları sunulur. CPC 464'ün komple bellek osiloskopunu değiştiren bu sistem ADU (Anolog-Dijital-Çevirici)'dur. Bu şüphesiz ki orijinal bir fikirdir, fakat 10 KHz'in üst frekanslarında kullanıcı için uygun değildir.

Becerikli amatörler şüpheli durumda CPC 464'ün çevre elemanlarından çekinmezler.

1.6 Kesim Konumu

Kesim konumu bilgisayar ve çevre donanımı arasında bir bağlantı kurmalıdır. CPC 464'ün arka tarafında farklı kullanımlara uygun birçok kesim konumu vardır. Aşağıdaki konu değişik kesim konumları sinyallerinin fonksiyon ve bağlantılarını açıklamaktadır. Ayrıca çevre birimlerinin hangi konuma ne şekilde bağlanabileceği de açıklanmaktadır.

1.6.1 Geliştirme Bağlantısı

CPC 464'ün geliştirme bağlantısı bilgisayarın yaklaşık tüm sinyallerini verebilmektedir. Bu kesim konumlarının bağlantısı gerçi CPC 464'ün el kitabında basılmışsa da bazı hususi sinyallerin anlamı belirtilmemiştir. Bunları Tablo 1.3'de bulabilirsiniz.

Hususi sinyaller aşağıdaki gruplara ayrılmıştır:

1. Veri (C0'dan D7'ye kadar)
2. Adresler (A0'dan A15'e kadar)
3. Yönetim kanalı (MREQ, M1, RFSH, IORQ, RD, WR, HALT, INT, NMI, BUSRQ, BUSAK, READY, RESET)

4. Bellek aktifleştirme sinyalleri (ROMEN, ROMDIS, RAMRD, RAMDIS)
5. Özel sinyaller (SOUND, CURSOR, LIGHTPEN, EXP, CLOCK)

İlk üç grubun sinyalleri CPC 464'ün mikro işlemcisi tarafından üretilir. ROM ve RAM belleğin aktifleşmesi ve aktifliğini yitirmesiyle ilgili sinyaller ile Gate Array uğraşır ve CLOCK hariç son grubun bütün sinyalleri PIO ve CRTC tarafından üretilir.

CLOCK sinyalini darbeyi dörde bölen Gate Array ve kristal üretir.

Normal olarak disketin durması geliştirme bağlantısına bağımlıdır; şayet bağlantının bütün çıkışlarını harekete geçirirseniz ve girişlerine TTL-seviyesinde sinyaller bağlarsanız, özel devrelerinizi de bağlayabilirsiniz.

Ayak No.	İşareti	Aktif durumdaki sinyalin anlamı
1	SOUND	Ton çıkışı
2	GND	Şase bağlantısı
3	A15	Adres kanalının en yüksek değerli Bit'i yerleştirilmiştir.
4	A14	Adres kanalının en düşük değerli Bit'i yerleştirilmiştir.
5	A13	" " "
6	A12	" " "
7	A11	" " " "
8	A10	" " "
9	A9	" " "
10	A8	" " "
11	A7	" " "
12	A6	" " "

Ayak No.	İşareti	Aktif durumdaki sinyalin anlamı
13	A5	" " "
14	A4	" " "
15	A3	" " "
16	A2	" " "
17	A1	" " "
18	A0	" " "
19	D7	Veri kanalının en yüksek değerli Bit'i yerleştirilmiştir.
20	D6	Veri kanalının en düşük değerli Bit'i yerleştirilmiştir.
21	D5	" " "
22	D4	" " "
23	D3	" " "
24	D2	" " "
25	D1	" " "
26	D0	" " "
27	+5V	Beslenme gerilimi
28	MREQ	Adres kanalı belleğe hükmeder
29	M1	İlk makina çevrimi başlar, yani CPU bellekten komut kodlarını alır.
30	RFSH	Adres kanalının alt 7 Bit'i dinamik RAM'ın refresh adreslerini kapsar
31	IORQ	Adres kanalının alt Bit'i I/O-Port'a kumanda eder
32	RD	Veriler okunur
33	WR	Veriler yazılır
34	HALT	CPU Bir sonraki kesime kadar bekler
35	INT	Cereyanı kesilen kesim talebi
36	NMI	Cereyanı kesilmeyen kesim talebi
37	BUSRQ	CPU'nun cereyanı kesilmistir ve çevre elemanı bellek ve I/O-Portlar da çalışır
38	BUSAK	Cereyanın kesilmiş olduğunun CPU tarafından belirtilmesi

Ayak No.	İşareti	Aktif durumdaki sinyallerin anlamı
39	READY	CPU çalıştırılır(gerilim uygulanmıştır)
40	BUS RESET	CPU'nun ön yüklenmesi (giris)
41	RESET	CPU'nun ön yüklenmesi (cıkıs)
42	ROMEN	ROM bellek calıştırır
43	ROMDIS	ROM bellek kapatır
44	RMRD	RAM bellek okuma icin acılır
45	RAMDIS	RAM bellek okumaya kapatılır
46	CURSOR	Kursör calıstırılır (CRTC'nin cıkışı)
47	L.PEN	Işık ibresi aktif (CRTC'nin girisi)
48	EXP	Gelistirme bağlanır (giriş)
49	GND	Şase bağlantısı
50	Φ	Darbe cıkısı

Tablo 1.3 Geliştirme bağlantısının ayak uçları

Şayet numaralar doğruysa ayrıca bütün sinyaller galvanik opto bağlayıcı üzerinde ayrılırlar. Böylece devrenize kısa devre yapması halinde hiç bir şey olmaz. +5 Volt'u 27 no.lu ayağa yerleştirir ve devrenizin şasesini 2 yada 49 no.lu ayağa bağlarsanız, geliştirme bağlantısında 1 Ampere kadar çıkabilirsiniz.

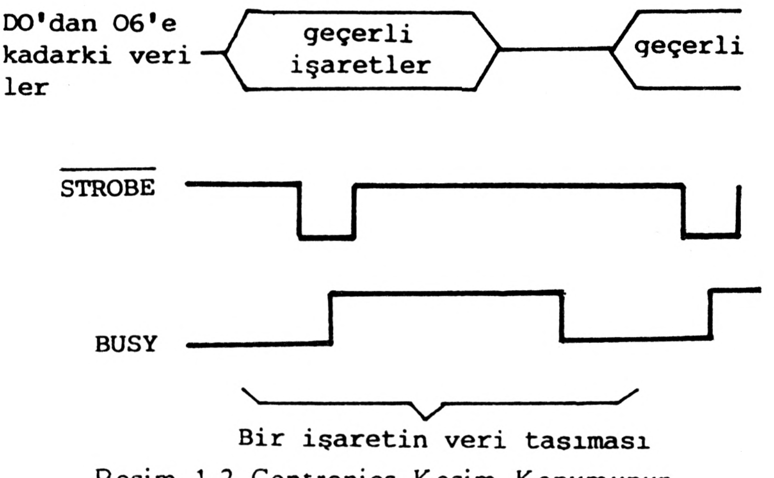
Buna rağmen dikkatli olmanız ve CPC 464'ün iletken yollarının bu akıma sürekli dayanıp dayanmayacağını kontrol etmelisiniz, zira pratik bilgi değerleri zamanımızda geçerli olmamaktadır.

1.6.2 Centronics-Kesim Konumu

Yapısı ve Fonksiyonu

CPC 464'ün basıcı bağlantısında Centronics-Kesim konumunun zayıf yorumu söz konusudur. Ayrıca standart kesim konumu olarak tarif edilen bütün sinyaller CPC 464 tarafından temin edilmez.

Centronics-Kesim konumunun CPC 464'ün sunmadığı en önemli sinyali en yüksek veri Bit'i D7'dir. Böylece CPC 464 ile 128'den 255'e kadarki ASCII değerleri bağlanan Basıcı'ya vermek mümkün değildir. Bunun için Basıcının satın alınması anında yazılımsal olarak Basıcı'dan en yüksek veri Bit'in yerleştirilip, yerleştirilmediğine dikkat ediniz.



Resim 1.2 Centronics-Kesim Konumunun zaman diyagramı

Aksi halde her Basıcı'nın 128'den 255'e kadarki ASCII değer alanında sunduğu özel çizimler ve grafik çizimleri basılamaz. Bu özelliğinden de görüldüğü gibi CPC 464'ün kesim konumu tamamen fonksiyoneldir. Karşılaştırma sinyalleri için $\overline{\text{STROBE}}$ ve BUSY sinyalleri bulunmaktadır, bu da kesim konumunun düzenli çalışması için yeterlidir.

Resim 1.2, karşılaştırma sinyalle Basıcının bağlı olduğu bilgisayar tarafından özel bir işaretin normal olarak aktarılmasını göstermektedir. Önce

CPC 464 D0'dan D7'ye kadarki veri bitlerini veri kanalına yerleştirir. Daha sonra basıcıya verileri alabileceğini belirten STROBE sinyali yerleştirilir. Basıcı verileri alır ve BUSY sinyaliyle meşgul olduğunu gösterir. Bu sinyal aktivitesini kaybedince bilgisayar basıcının serbest olduğunu ve daha sonraki işareti taşıyabileceğini bilir.

Basıcının Kablosu

Basıcının CPC 464'e bağlanması için, bilgisayar tarafı 34 kutuplu direkt bağlantı fişi, Basıcı tarafı normal Centronics-fişli özel bir kablo gerekmektedir. Bu kablo malesef pek ucuz değildir. Biraz bilginiz ve beceriniz varsa, kablonun birkaç metre yüzey kaplamalı kablo ve iki fiş ile yapabilirsiniz.

Tarihçe

Centronics-Kesim konumu ismini, bu kesim konumlu basıcıyı ilk kez üreten aynı isimli Basıcı firmasından almaktadır. Diğer üreticiler bunu takip etmiş ve pratik sebeplerden dolayı aynı kesim konumunu kullanmışlardır. Kısa sürede standartlaşmış ve bundan sonra her Basıcı'da kesin konumu bulmuştur.

Fakat daha sonra her Basıcı üreten firma kendi "Centronics-Standart'ını" geliştirmiş ve üretmiştir. Bundan dolayı kesim konumlu yapılan bir bilgisayarı Centronics-Kesim konumlu bir Basıcıya bağlamak bugün için kısmen mümkün değildir. Dikkat: Her Centronics aynı değildir.

Bundan dolayı Basıcınızı ve basıcı kablonuzu alırken cihazınızın bağlantısındaki ayak uçlarına uygun olup olmadığına dikkat ediniz. Aksi halde basıcınız çalışmaz ve CPC 464'ünüzdeki bazı entegre devreler

(IC-elemanları), yanar! (Basıcının bağlanması anında herhangi bir kesim konumu problemi olmamalıdır.)

1.6.3 Joystick İçin Bağlantı

Joystick bağlantının ayak ucu genel standarta uygundur, yani her fabrikanın ürettiği joystick CPC 464'e bağlanabilir. Bunun için bilgisayarınızın sadece bir joystick bağlantısının olduğuna dikkat edilmelidir. Fakat bununla beraber iki oyuncunun aynı anda karşılıklı direkt mücadelesini mümkün kılan birçok oyun vardır. Bu nasıl olmaktadır?

Sizin sadece bir Joystick'iniz bulunmaktadır. Bu Joystick tek özellik ise, üzerinde diğer bir Joystick'in bağlanabildiği bir Joystick yuvasının bulunmasıdır. Şayet oyunlarınız ve diğer programlarınız için sadece bir Joystick'e ihtiyacınız varsa yada Joystick'inize ikinci bir Joystick bağlamak istiyorsanız, arzu ettiğiniz normal herhangi bir marka Joystick kullanabilirsiniz.

Böylece en azından Joystick satın almaktan kurtulursunuz, tabii Joystickli bir tele oyun yada eski bilgisayarınızın Joysticklerini kullanmak isterseniz.

Common ve COM2 Bağlantı Ayakları

Belki bilgisayarın bir Joystick bağlantı üzerinde iki Joystick'i nasıl ayırdedebildiğini kendi kendinize sorabilirsiniz. Çözümü oldukça basittir: Bilgisayar Joystick bağlantısına COMMON ve COM2 sinyallerinin her ikisini de verir. COMMON sinyali aktifleşince ilk Joystick, COM2 sinyali aktifleşince ikinci Joystick soruşturulur. Her iki sinyal COMMON ve COM2 her iki Joystick'in adreslenmesine yarar.

Teorik olarak CPC 464'e istediğiniz kadar Schneider marka Joystick bağlayabilirsiniz, zira iki mevcut COM bağlantı ayağının yanında başka hiç bir COM bağlantı ayağı yoktur, aksi halde ilâve olarak bağlanan Joystick'lerin soruşturulması bilgisayar için mümkün olmazdı.

1.6.4 Diğer Kesim Konumları

Diğer kesim konumları olarak CPC 464'de; monitörün bağlanması için bir video çıkışı ve bilgisayarın stereo-tertibata bağlanması için de bir stereo-ton çıkışı vardır.

Video çıkışın bağlantı ayakları CPC 464'ün el kitabında belirtilmiştir ve bizim için pek enteresan değildir. CPC 464'ün Stereo-Ton çıkışı size, bilgisayarınızın hoperlöründen kısıt ve donuk mono bir ton yerine Hİ-Fİ cihazınızın tam ve dolu stereo tonlarını sunmaktadır. Bunun için sadece ton çıkışına 3,5 mm'lik mandallı normal bir kablo bağlanmalı ve kablonun diğer tarafı stereo-tertibatınızın MONITOR yada AUX-girişine bağlanmalıdır.

2

CPC 464'ÜN BELLENİM SİSTEMİ

Bellenim sistemi, donanım ve yazılım sistemi arasındadır, aynı donanım gibi üretici tarafından bilgisayar ile beraber gönderilir ve bilgisayarı yöneten programlardan (yazılım) oluşur.

Bellenim, cihazın satışı sırasından sabitleştirilip takılmış yada beraberinde gönderilmiş tüm yazılım sistemi olarak belirtilebilir.

CPC 464, bilgisayarın komple çalışma sistemini içeren 32 KBit büyüklüğünde ROM salt okunur belleğe haizdir. Bu ROM bellekte bir BASIC-yorumlayıcı, Tuş takımı, Ekran, ses üreteci, kaset-teyp ve sadece önemli sistem programlarını belirtmek için satır ve ekran yönetim Editörü bulunmaktadır.

Bu bölümde, CPC 464'ün belleniminin anlaşılıp kullanılmasından söz edilmektedir.

2.1 Tuş Takımı Fonksiyonları

CPC 464'ün tuş takımı Amerikan daktilo klavyesinden ve sınırlı rakamlardan oluşmaktadır. Bütün fonksiyon tuşları renkli olarak ayrılmış, yalnız CLR-tuşu, DEL-tuşu ile karışmaması için bunlardan ayrı tutulmuştur.

Özel tuşların fonksiyonları CPC 464'ün el kitabında geniş bir şekilde anlatılmıştır, burada ise sadece kısa olarak işlenip genişletilmiştir.

Programın uygulanması ve listeleme sırasında ESC-tuşuna bir kez basmakla bir duraklama oluşur, iki kez basmakla faaliyet durur.

DEL-tuşu kursör'ün (ışıklı göstergenin) solundaki işareti siler, CLR tuşu ise kursör pozisyonunun geçici işaretini siler.

COPY tuşu ile kopya kursör pozisyonunun işareti munzatam kursör pozisyonuna kopya edilir.

CAPSLOCK tuşuna basmakla bütün mevcut harfler büyük harf olarak oluşur, CAPSLOCK tuşuna yeniden basılması bu fonksiyonun tersini oluşturur.

CTRL ve CAPSLOK kendi fonksiyonlarında yazı makinalarının SHIFTLOCK Fonksiyonuna uymaktadır. Dikkat, kopya kürsoru kürsor tuşu ile hareket ettiğinden bu durumda Editör kopya kürsoru ile çalışmaz.

CTRL ve diğer bir tuşa basmak suretiyle 1'den 31'e kadar ASCII işaretleri oluşur, CTRL M; Enter tuşuna uyar ve CTRL P ince bir ses tonu üretir.

TAB tuşu herhangi bir tabulatör fonksiyonu değil, aksine sağ tarafı gösteren bir ok oluşturur.

ENTER tuşunu açıklamaya gerek yoktur, zira bu tuş bilgisayarın en önemli ve en çok kullanılan fonksiyonudur.

SHIFT CTRL ESC'nin anlamı sizin için oldukça önemlidir. SHIFT ve CTRL tuşuna aynı zamanda ve daha sonra da ESC tuşuna basmakla CPC 464'ün yazılımında RESET durumu oluşur. Bütün bilgiler silinir ve bilgisayara yeniden ön yükleme yapılır.

Sonuç olarak birkaç sözde | (SHIFT @): işaretinden söz edelim.

Bu işareti bir REM satırına yerleştirirsek, ol-

dukça tuhaf şeyler oluşur. Aşağıdaki satırları kaydediniz ve bu satırları CPC 464'e listeletiniz:

10 REM | !

Ünlem işaretini kaybolur gibi göreceksiniz. Bu ne program metninde bulunmakta, ne de | işareti aşağıdaki satırların listelenmesini önlemektedir.

İlaveten şu satırları kaydediniz ve programı çalıştırınız:

20 PRINT"HALLO!"

Yine şaşıracaksınız, zira CPC 464 hiç bir şey belirtmeyecektir. Program satırı 20 listelenebilir ama bu satırın uygulanması daha önceki REM-satırındaki | işareti ile engellenir.

Denemeler suretiyle bu işaretin diğer özelliklerini de tespit edebilirsiniz. Düzenli bir program işlevinde | işareti REM satırında kullanılmaz.

CPC 464'de Umlaut'ları belirtmek istiyorsak, işaretinden küçük ö harfi oluşur. ASCII-değeri değişmediğinden, bu işaretin herhangi bir REM-satırına yerleştirilememesi, ö harfi için de geçerlidir. Bunun için soruşturma programının REM-satırlarına ve Adres yönetiminin REM satırlarına bakınız!

2.2 Editör

CPC 464'ün Editörü, yazı hatalarının düzeltilmesi yada bir BASIC program satırının düzeltilmesi hususunda size birçok imkânlar sağlamaktadır. Bir hatayı ortadan kaldırmak için toplam dört metod vardır:

- o Yanlış bir kayıt işlemi kürsor tuşunun ve CLR ve DEL tuşlarının kullanılması ile ortadan kaldırılabilir.
- o Hatalı bir program satırını, satırı yeniden kaydetmek suretiyle düzeltebilirsiniz.
- o EDIT komutu ile düzeltilmesi gereken satıra atlanabilir ve hata işaretin ilâvesi yada silinmesi ile düzeltilebilir.
- o Kopya kürsorü, satırların ve ekran ile aynı zamanda CPC 464'ün belleğinde satır aralıklarının istenilen yere kopya edilmesini sağlar.

En hızlı editör metodu hatalara göre farklılık gösterir.

Basit kayıt hataları ilk metodla derhal düzeltilebilir. Çok kısa olan satırları yeniden kaydetmek en iyisidir. Hataları satır başlarında bulunan program satırları en hızlı bir şekilde EDIT komutu ile düzeltilebilir, zira bu durumda bütün satırı kursör ile geçmeye gerek yoktur.

Aynı şekilde az hatalı program satırları, bu durumda satırları kursör ile enine ve boyuna geçebileceğiniz için, EDIT komutu ile düzeltilebilir.

Program satırlarının birleştirilmesi kopya kursörü ile oluşmaktadır, zira EDIT-modunda aktuel olarak edite edilmiş satırlar ayrılamazlar.

Bir satırın daha fazla sütunundaki daha büyük hatalarda bu metod görülebilmesi açısından en uygun metoddur. Sadece COPY tuşu ile kopya edilen ve yeniden tuşlarla kaydedilen işaretler ekranda görülebilir ve yeni program satırının içeriği olarak kaydedilirler.

Pratik uygulamalarda hangi durumlar için hangi editör imkânlarının en hızlı olduğu zamanla öğrenilebilinir. Her metodun faydaları ve mahsurları var-

dır. CPC 464 ile hataların düzeltilmesi diğer bilgisayarlardan daha rahat ve kolaydır.

2.3 BASIC-Yorumlayıcı

BASIC-yorumlayıcı bellenim sisteminin en büyük ve aynı zamanda en önemli sistem programıdır. Bu program özel BASIC-komutlarını yorumlar, onları bilgisayar kodlarına çevirir ve (mümkün olduğu kadar) uygular. BASIC-yorumlayıcı esasen BASIC programlama dilini "0" ve "1" sayılarından oluşan ve bilgisayarın anlayabileceği sayı kodlarına çeviren bir çeviri programıdır.

BASIC-yorumlayıcı komutları ne kadar iyi anlar ve işleyebilirse, programlayıcı için o kadar verimli olur. BASIC-yorumlayıcı 170 komuta ve böylece ev bilgisayarlarının BASIC-Dialektlerine haizdir.

2.3.1 Değişkenler

Değişken İsimler

CPC 464'ün değişkenlerinin isimleri 40 satıra kadar uzanabilir ve alfanümerik işaretlerden oluşur. Değişken isimlerin bir harf ile başlaması ve hiç bir BASIC-komut kelimesine uymaması (meselâ END gibi) özel şarttır. Bir değişken ismin bölümü olan BASIC-komut kelimesi ise (meselâ ende) kullanılabilir. Komutlar ve değişkenler (yada sayılar) arasında daima bir boşluğun bulunması gerekir.

Diğer taraftan BASIC-yorumlayıcı bütün ifadeyi değişken olarak yorumlayabilir.

Şayet Alman dilindeki Umlaut'ları belirtmek istersek, bu Umlaut'lar değişken isimlerde bulunamaz, çünkü bunlar esas işaretlerin ASCII değerlerine

sahiptirler ve bu işaretler yorumlayıcı tarafından harf olarak algılanmazlar.

Değişken Tipleri

CPC 464'ün üç değişik değişken tipi vardır:

Reel değişkenler -1,70141E+38'den +1,70141E+38'e kadar uzanan bir alanda herhangi bir değer alabilirler. Bu değişken üstel yazı tarzında belirtilmiyorsa dokuz rakam, üstel yazı tarzında ise sadece altı rakam görülür. Reel değişken DEFREAL komutuyla yada değişken ismin arkasından hemen bir ünlem işareti ile tanımlanabilir. -32768'den +32767'ye kadar uzanan alandaki bir tam sayı değeri tam sayı değişkeni olarak belirtilebilir. Bu sayı alanı 16-Bit yani 2-Bit'lik ifade ile tanımlanır.

Bir tam sayı değişkeni DEFINT komutuyla yada değişken ismin ardından bir yüzde işaretinin kaydedilmesi ile tanımlanabilir.

Dizgi değişkenleri, CHR$(32)'den CHR$(255)'e kadar uzanan ASCII işaretlerinden terkip olan 255 karakter uzunluğundaki bir metni içerirler. DEFSTR komutu ile yada değişken isimden sonra bir dolar işareti verilmesi ile bu değişken tip tanımlanabilir. Şayet bir değişken tipi DEF komutu ile yada değişken isimden sonra özel bir işaret verilmek suretiyle tanımlanmıyorsa, yorumlayıcı tip olarak otomatikman REAL durumunu alacaktır.

Değişkenlerin Belleğe Kaydedilmesi

Şayet değişkene bir değer veriliyorsa yada değişkenin içeriği değişecekse bu yeni değer BASIC-yorumlayıcı tarafından belleğe kaydedilmelidir. Bu

amaç için CPC 464'ün çalışma (iş) belleğinde Program metninden hemen sonra bir değişken tabelası mevcuttur.

Programlama sırasında komutların ilâve edilmesi ve silinmesi ile program metninin son pozisyonu daima değişir. Birçok bilgisayarda bunun sonucu olarak değişken tabelaları tahrip olur yada tüm değerler silinir.

CPC 464 program değiştirilmesi sırasında, tabelaları belleğinde uygun şekilde sıralayarak, bu değişkenlere dikkat eder.

Şayet programınızı RUN komutu ile başlatırsanız, otomatik olarak bütün değişkenler silinir. GOTO komutu ile de bu durum önlenebilir. Böylece bütün değişken değerler olduğu gibi kalır ve program atlanan satırda çalışmasına başlar. Fakat bu satırda bir CLEAR komutu bulunursa, mevcut bilgiler silinirler.

2.3.2 BASIC-Komutları

CALL

CALL komutu ile bellenim sisteminin özel RAM yada ROM yordamları (programları) çağrılabilir. Bu komutu kullanırsanız ve parametrik (değişken) nicelik kaydederseniz, bütün değerler -32768 ile 65535 arasındaki alanda bulunurlar.

CHR$

CHR$+CHR$(n) komutu n sayısının ASCII değerini belirtir. Bu şekilde ASCII'yi 1'den 31'e kadar getirmek mümkündür.

CHR$(7) kaydedildiği zaman bilgisayar ince bir sesle cevap verir.

CHR$(22)+CHR$(1) komutu ile transparent modu çalışır, CHR$(22)+CHR$(0) komutu ile bunu tekrar kaldırabilirsiniz.

CHR$(24) komutu kâğıt rengini yazı rengi ile değiştirebilirsiniz. Böylece siyah-beyaz monitörün sahibi olarak video resmi invertleyebilirsiniz.

DEC$

DEC$ komutu bilgisayarda BASIC-Kodları gerçekleştirilmiştir. Burada dec$ yazımının diğer BASIC komutlarında olduğu gibi listeleme sırasında otomatikman büyük yazım şekli DEC$'e çevrildiğini göreceksiniz. Her deneme sırasında bir hata belirtisi görüldüğü için bu komutu yönetmek mümkün değildir. BIN$ ve HEX$ komutlarına eşdeğer olan DEC$ komutu bir sayıyı desimal sayı sistemine çevirir. İkili ve altılı sayılar PRINT komutu ile otomatik olarak desimal sayılara çevrildiği için bu fonksiyon lüzumsuzdur.

ENT

Konu 2.5.4'e bakınız.

ENV

Konu 2.5.3'e bakınız.

IN KEY$

Bir veri programa ara verilmeden INKEY$ komutu kaydedilebilir.

Bazen INKEY$ komutu ile INPUT komutunun yanlışını göstermek mümkündür. Böylece bilgisayar tuşları soruşturmak suretiyle bir verinin kaydedilmesiyle paralel olarak diğer işlemleri de gerçekleştirebilir. Eğer programının CALL 47944'le program kesilmesine karşı korumak istiyorsanız bir imkân daha ortaya çıkmış olur. Her INPUT komutu bu fonksiyonu sona erdirir. INKEYS komutunun kullanılması ile aksaklık önlenmiş olur.

INPUT'la basit bir yanlış gösterme

```
10 text$=" "
20 FOR lauf=1 TO n
30 a$=INKEY$:IF=" " THEN 30
40 text$=text$+a$
50 next lauf
(lauf yerine bir değişken konulacaktır)
```

Bu küçük program n işaretinden oluşan bir metnin kaydedilmesini sağlar. Böylece CLR ve REL tuşlarının editör ihtimalleri işlev dışı kalacaktır.

Merge

MERGE komutu ile birçok özel BASIC-programlarını büyük bir programla birleştirmek mümkündür. Tek şart; yeni program satırlarının eski satırların üzerine yazılmaması için programın satır numaralarının çatıştırılmamasıdır. Eğer RENUM komutu ile programın satır numaralarını birleştirmeden önce numaraları değiştirirseniz hiç bir yanlışlık olmaz.

MOD ve

Mod ve komutları CPC 464'ün el kitabında gösterilmemiştir, buna rağmen cihaz üzerinde kusursuz bir şekilde çalışırlar. Eğer bir bölme işleminde birleştirme işareti "/"nin yerine " " işareti kullanırsanız bölme işleminin tam sayılı sonucunu elde edersiniz. Aynı durumda MOD komutu yerleştirilirse, bölmenin tam sayı kalanı bulunur.

OPENIN ve OPENOUT

Şayet bir veriyi OPENOUT name komutu ile name ismi altında hafızaya yüklerseniz, bu veri OPENIN name komutu ile tekrar okutulabilir. Aksi halde hatalı olacaktır. Veri isimleri sadece büyük harflerden oluşabilirler, yani veri name ismi altında değil, bilâkis NAME ismi altında geçirilir. Şayet bu veriyi name ismi altında okutmak istiyorsanız, name ve NAME bilgisayar için aynı şeyi ifade etmediğinden dolayı bilgisayar saçmalamaya başlayacaktır.

Verileriniz sabit bir isim değil de, değişken bir isimse bu durumda daha fazla karmaşıklık oluşacaktır.

OPENOUT name$ komutu ile bir veriyi hafızaya kaydederseniz ve name $ değişkeni küçük harfle olursa, bu veri yalnız OPENIN UPPER $ (name $) komut zinciri ile okutulabilir. Name $ değişkeninin metin uzunluğu değişik şekillerde oluşuyorsa INPUT name $ metninden önce name $ = STRING$(16,") komutu verilmelidir. Bu komut değişkene 16 boş işaretten (karakterden) oluşan bir metin sağlayacaktır. Şimdi haklı olarak bunun neden yapılması gerektiğini soracaksınız.

Şayet OPENOUT name $ komutu ile bir veriyi

hafızaya kaydetmek isterseniz ve name $ değişken metninin uzunluğu her defasında daha da artacaksa, kaydedilmesi gereken veri belki de name $ değişkenini almayacaktır. Bellenim sisteminin bu hatası, gerçek metninizin yazılmasından önce değişkene maksimal (veri isimlerinde 16 yer) uzunlukta bir metin ayrılması ile ortadan kaldırılabilir.

Bu durumu anlayamadıysanız, 3 bölümdeki soruşturma programında verilerin kaydedilmesi ve okutulması durumuna bakınız.

RANDOMIZE

RAMDOMIZE komutuna, rastgele bir sayı zincirinin başlangıcını ifade eden bir sayı kaydedilmelidir. RANDOMIZE komutuna TIME kaydederseniz, her program işleyişinde diğer bir rastgele sayı zinciri oluşacaktır, zira TIME değişkeni RANDOMIZE komutunun her geçişinde değeri bir değeri kapsayacaktır.

RENUM

RENUM komutu ile programınızın satırlarını istediğiniz şekilde numaralayabilirsiniz. Bununla beraber başlangıç noktasının yanında numaralama işlemi için bir de bitiş noktası tespit etmek mümkün değildir. Programınızı daima baştan sona kadar numaralatmalısınız. Kendi sıralarında bulunan esas programları bu programda yeniden numaralarsanız amaca ulaşmış olursunuz, çünkü RENUM komutu başlangıç noktasından itibaren daha önceki RENUM komutunun numaralamasını mantıki bir şekilde ortadan kaldırır.

Böylece, konu 2.3.3'de de belirtildiği gibi, bir

tasnif işlemini gerçekleştirebilirsiniz.

RND

RND komutu kullanılan BASIC-diyalektlerinin alışkın olduğu şekilde çalışmaz. Minimumdan maksimuma kadar rastgele bir sayı üretmek için INT (RND (maximum-minimum+l)+minimum) ifadesini kullanmanız gerekmektedir.

SOUND

Konu 2.5.2'ye bakınız.

SYMBOL

SYMBOL komutu CPC 464'ün el kitabında pek anlatılmamıştır. Şayet yeni bir sayı ifade etmek istiyorsanız. bilgisayara önce SYMBOL AFTER n komutu ile ASCII işaretlerinin nokta matrislerinin n sayısından itibaren RAM hafızaya kopya edilmesinin gerektiği bildirilmelidir. Şimdi SYMBOL komutu ile n'den 255'e kadarki bütün ASCII değerlerini değiştirebilirsiniz. Şayet n sayısı 239'dan büyükse, SYMBOL AFTER komutu kullanılmayabilinir, zira 240'dan 255'e kadarki ASCII işaretlerinin nokta matrisleri standart olarak RAM hafızada bulunurlar ve bundan dolayı hafızaya kopya işlemi gereksizdir.

SYMBOL komutunun ilk parametresi yeniden ifade edilmesi gereken işaretin ASCII değerini belirtir. Diğer sekiz parametre yeni sembol için Bit (ikil) örneğidir. İlk değer yerleştiği ikiller suretiyle (Bit), yeni işaretin üst satırında hangi noktaların yerleştirilmesi gerektiğini belirtir, son değer alt satırların örneğini temin eder.

Örnek: SYMBOL 240, 255, 129, 129, 129, 129, 129, 129, 255 komutu 240'ın ASCII değeri için bir dikdörtgeni ifade eder. 255'in ikili değeri 11111111'dir, yani 240 işaretinin birinci ve sekizinci satırı tamamen yerleşmiştir ve dikdörtgeni alt ve üst tarafını teşkil eder. 129'un ikili değeri 10000001'dir. Yani birinci ve sekizinci satır arasındaki bütün satırlarda sadece birinci ve sonuncu satır doludur. İkiden yediye kadar olan satırlar dikdörtgenin yanal sınırlarını (yan taraflarını) oluşturur. Bütün satırların biraraya getirilmesi ile tam bir dikdörtgen oluşur. Bu durumun öğrenilebilmesi için birçok denemenin yapılmasında fayda vardır.

TRON ve TROFF

TRON ve TROFF komutları programdaki hataların mükemmel bir şekilde aranmasına yararlar. TRON komutuyla Trace-Modunu çalıştırırsanız, bilgisayar çalışan program satırlarının satır numaralarını ifade eder. Böylece programın ve hangi satırların çalıştığını görebilirsiniz. TROFF komutu ile Trace-Modunun çalışmasını kesebilirsiniz.

ZONE

ZONE komutu ile baskı alanının genişliğini tespit edebilirsiniz. PRINT kumandası ile virgülle ayrılmış verileri kaydedersek, bu baskı alanları cevap verebilirler. İlk virgülün önündeki veriler ilk baskı alanına kaydedilir, ilk virgülden sonraki veriler ikinci baskı alanına kaydedilir.

Baskı alanı ekranın arta kalan kısmından daha genişse, bu alan otomatikman bir sonraki satırın başına yerleştirilir. Böylece örneğin MODE 1'de ve

ZONE 41'de bütün veriler, tek bir alana bir satırın uyduğundan dolayı, altalta kaydedilirler.

Tabela hazırlamak için ZONE komutu özellikle çok uygundur.

Sayıların Yuvarlatılarak İfade Edilebilmesini Sağlayan Komutlar

Sayıların yuvarlatılması için dört komut bulunmaktadır: CINT, FIX, INT ve ROUND.

CINT komutu, sayının virgülden sonraki kısmı en azından 0,5 ise, sayıyı tam alanında yuvarlar (-32768'den 32767'ye). Negatif sayılar da yeküne bağımlı olarak yuvarlatılır, yani sayılar -'ye değil aksine bir üst sayıya yuvarlatılır!

FIX komutu herhangi bir sayının sadece virgülden önceki kısmını verir, INT komutu bir sayının en küçük tam sayılı değerini ifade eder, böylece her ikisi de sayıları yuvarlatmış olur.

ROUND komut ile sayılar virgülden sonraki sabit bir değere yuvarlatılır.

Arzu ettiğiniz yuvarlatma tarzına göre uygun komutlardan birini kullanabilirsiniz.

ROUND fonksiyonu INT (zahl * 1En+0,5)/1En komut zincirini uygun şekilde tamamlar. Bu fonksiyon zahl değişkenini virgülden sonraki n yerine yuvarlatır, örneğin: INT(123.456 * 100+0,5)/100 sonuç olarak 123,46 sayısını verir.

Giriş ve Çıkış Birimleri

CPC 464 #0'den #9'a kadar onlu giriş-çıkışa sahiptir. #0'dan #7'ye kadar olan birimler ekrandan n

adet kumanda PRINT #n ekranda n adet bılgi verir. INPUT #n, emri ekranda (monitörde) (n) adet bilgi girişini gerektirir.

#8 Birimi Centronics-kesim yeri üzerinden yazıcıyı uyarır.Bu birim ile sadece verileri belirtebilirsiniz. Şayet CPC 464 üzerinde yazıcı bağlı durumda değil ise bilgisayar yazıcı ikazını bekler ve ESC tuşuna iki kez basılması suretiyle kesilir.

#9 Birimi kaset-teyp'i uyarır.Verileri girmek ya da okumak için,önce giriş ve çıkış verilerini açmak verileri INPUT#9 ya da PRINT#9 ile kaydetmeli ya da okumalı ve bunları müteakip verileri tekrar kapamalısınız.

Verilerin kaydedilmesi ya da okunması anında herhangi bir birim belirtilmemiş ise, CPC 464 tarafından otomatik olarak #0 birimi kullanılır.

2.3.3 Yapısal Programlama

Bu arada son BASIC-programlayıcılarda Spaphetticode-programı olarak adlandırılan program tarzının daha zayıf olduğuna dikkat edilmiştir. CPC 464 yapısal programlamayı koruyan daha fazla özelliklere sahiptir.

Değişken isimlerinin muhtemel uzunlukları programlayıcıda, programın değişebilen büyüklükleri için onları ifade eden isimlerin kullanılmasını mümkün kılar.REM komutu yada "'"işareti ile bazı bölümlerin fonksiyon dökümanlarını programınızın yorumlama satırlarına ilave edebilirsiniz, böylece bunlar haftalarca programınızda kalabilir.

Program içersinde en az iki defa kullanılan program kesimleri, alt programlar olarak belirtilse,

görülebilir programlar elde edebilirsiniz. Diğer program kesimleri giriş ve çıkışlar halinde tek tek program bloklarında (GOTO-atlamaları olmaksızın) bölünebilir ve tasnif edilebilir.

Pratikte, bilinen satır numara alanlarına belirli fonksiyonlar verilmiştir. Örnek olarak:

1'dem 9'a kadar REM programlayıcı,kopyalama v.b
10'dan 99'a kadar Değişkelerin, renklerin belirtilmesi
100'den 299'a kadar Program listeleri
300'den 999'a kadar Program blokları(ana programlar)
1000'den 9999'a kadar Alt programlar (Subroutinen)
10000'den 59999'a kadar çapraşık alt programlar
60000'den 65535'e kadar program bilgileri

Bu şema tabii ki her şeyi açıklamaz, ama kendi bölümlerinin düzenini belirtir.

2.3.4 Programda Hata Arama

Tarafınızdan hazırlanan programların uzunluğu birkaç satırı geçtiğinde gayri ihtiyari hatalar oluşacaktır. Programda hata oluştuğunda bu hatanın bir yazı hatası mı yoksa bir düşünce hatası mı olduğuna karar verilmelidir.

Bilgisayar yanlış program satırını hata göstergesi ile belirttiği için, yazı hatalarının bulunması pek zor olmaz.

Düşünce hatalarının bulunması ise bunun aksine oldukça zordur. Hata arama işlemini kolaylaştırmak

için kritik yerlerdeki önemli değişken değerlerini sürülen PRINT-kumandası ile bastırabilirsiniz. Bu ölçü size daha fazla yardım edemiyorsa, programın doğru işlemesini engelleyen bazı bilgileri üretilen değer verileriyle belirtiniz.

"RUN başlangıç" (bütün değişkenleri siler) yada "GOTO başlangıç" (hiç bir değişkeni silmez) ve "Son ENDE" komutuyla "başlangıç" ve "son" satırları arasındaki programda hatayı kavrayabilirsiniz. Bu alan ne kadar küçükse, hatayı lokalize etmek okadar basit olur.

Sonuç olarak programın beklediğimiz şekilde çalışıp çalışmadığını gösteren TRON ve TROFF komutları oluşurlar.

Bu metodla hatayı bulamazsanız, Programı belleğe yüklemeniz ve birkaç gün sonra yeniden hata aramanız gerekir. Problem oldukça derin olarak incelendiğinden dolayı hataların sık sık ortaya çıkması normaldir. Diğer- bir imkân ise, BASIC-bilen bir arkadaşınızın bulunup bulunmayacağıdır. Problem yerinin açıklanması hatayı ortaya çıkarabilir.

2.4. Grafik

CPC 464'ün grafiği oldukça güçlü ve verimlidir. 27 mevcut renkten 16 renge kadar aynı anda tanımlanabilir. Size iki renk yeterli oluyorsa, 128.000 resim noktası belirtilebilir.

CPC 464'de farklı birçok renk ve resim noktasını hazırlayan üç ekran mod'u vardır.

Mod 0 "çok renk modu" olarak tanımlanır. Her satır 20 işaret ve 160 x 200 resim noktası ile tarif edilir, bunun için aynı anda 16 değişik renk mümkündür. Mod 1 "standart mod"dur. Her satır 40 işaret ve

320 x 200 resim noktası ile gösterilir ve maximal dört değişik renk kullanılabilir. CPC 464'ün yüksek bölünen Modu ise Mod 2'dir. Her satır 80 işaret ve 640 x200 resim noktası ile tarif edilir ve sadece iki renk mümkündür.

Birçok bölünme imkânları olmasına rağmen, BASIC-komutlarıyla noktaların verilmesi anında modlar arasında hiç bir fark yapılmamalıdır.

Koordinatların ekran haricinde bulunması anında herhangi bir hata ikazı oluşmaz, bilâkis sadece noktalar yerleştirilmez. Bu ekran çerçevesinde bulunan grafiklerin verilmesi için oldukça faydalıdır.

2.4.1 Bölünme

CPC 464 yüksek grafik bölünmesi mükemmel fonksiyon grafiklerinin ve boyutlu ifadelerin gerçekleşmesini mümkün kılmaktadır.

Matematiksel fonksiyonların teşkil edilmesi için ORIGIN-komutu kullanılır. Bu komutla grafik koordinat sisteminin sıfır noktası istenildiği gibi yerleştirilebilir. Fonksiyon grafiğinin koordinat sıfır noktasına bu sıfır noktasını yerleştirirseniz, fonksiyon noktasının relatif kayması hesaplanmaz, çünkü negatif değerlerde tarif edilebilir.

Resim noktaları PLOT komutuyla yerleştirilebilir ve çizgiler DRAW-komutuyla çizilebilir. Elips ve çember çizimleri için CIRCLE komutu yoktur. Bölüm 2.4.5'de bunu nasıl kumanda edebileceğiniz gösterilmiştir.

Bazen yazı ve grafiğin aynı anda ekrana getirilmesi gerekmektedir.

Bu durumda metin grafiğin üzerine yazılır yada metin ve grafik kesişirle.. Ayrıca bu konuda CHR$(22) + CHR$(1) komutuyla çalıştırılıp, CHR$(22) + CHR$(0) komutu ile bitirilebilen transparent modu vardır.

Belki, CPC 464'ün yazılım sisteminin niçin 400 dikey resim noktasını belirtip, bilgisayarın donanım sisteminin sütun başına 200 noktayı hazırladığını kendi kendinize sorarsınız. Bunu açıklamak için biraz daha incelemek gerekir:

CPC 464 mod 0'dan toplam 32.000 resim noktasını tanımlar. Her resim noktası 16 renk alır ve dört Bit'le tam 16 durum tarif edilebildiği için bellekte 4 Bit'lik yer gerektirir. Her nokta 32.000 resim noktasından tam 4 Bit kaplarsa ekran belleği için toplam 128.000 Bit gerekir.

CPC 464 mod 1'de toplam 64.000 resim noktasını tanımlar. Bu durumda resim noktaları 4 renk alırlar ve 2 Bit'lik yer kaplarlar. Böylece ekran belleği için 128.000 Bit'e gerek duyulur.

Nihayet mod 2'de 128.000'lik bellek alanına gerek duyulur.

Ekran belleği için her durumda 128.000 Bit'in kullanıldığını görüyorsunuz. Bu 128.000 Bit CPC 464'ün video resmi için ayrılan üst 16 KBayt'lık RAM belleğine uygun düşer.

Şimdi böylece 400 resim noktasının durumu kolayca açıklanabilir. CPC 464'ün grafiğinde sütun başına 200 değil bilakis 400 dikey resim noktası bulunursa her durumda 256.000 Bit kullanılacağından ekran belleği için iki kat fazla bellek alanı hazırlanmalıdır. Sonuç olarak programlayıcıya RAM bellekte 43,5 KBayt değil aksine 27,5 KBayt ayrılmıştır.

CPC 464'ün BASIC'i toplam 256.000 resim noktasını belirttiği için, yakın bir zamanda bilgisayarın çalışma belleği küçülmeksizin grafik bölümünü iki katına çıkartan iç RAM bellek genişletilmesi tavsiye edileceği beklenmelidir. 256.000 resim noktasının en yüksek bölünmesi kullanılıcıları mutlaka memnun edecektir!

2.4.2 Renkler

CPC 464'ün renkleri sayılarla kodlanmıştır. Bu sayılar renk değerleri olarak tanımlanırlar.

PEN komutu ile yazı rengi, PAPER komutu ile arka plân rengi, BORDER komutu ile kenar rengi tespit edilebilir. Sadece BORDER komutunda renk değeri direkt verilebilir. Diğer komutlarda renk listesi numaraları belirtilmelidir.

CPC 464'ün grafik çözümü ne kadar yüksekse, renk listesi o kadar az kullanılır ve renkler ekranda aynı zamanda o kadar az belirtilebilir. Renk listesi içeriğinin değiştirilmesi ile özel renklerin birbirleriyle değişmesi mümkün olur.

CPC 464'ün renkleri ve renk değerleri aşağıdaki gruplara ayrılır:

Renkler:	Kırmızı Renkler:	Sarı Renkler :	Yeşil Renkler:
	16 Gül rengi	25 Pastel sarısı	22 Pastel yeşili
	7 Erguvan rengi	24 Açık sarı	21 Limon yeşili
	6 Açık kırmızı	15 Turuncu	19 Deniz yeşili
	3 Kırmızı	12 Sarı	18 Açık yeşil
			9 Yeşil
	Firuze Renkler:	Mavi Renkler:	Menekşe Renkleri:
	23 Pastel firuze	14 Pastel mavi	17 Pastel magenta
	20 Açık firuze	11 Gök mavisi	8 Açık magenta
	10 Firuze	2 Açık mavi	5 Açık menekşe
		1 Mavi	4 Magenta
Renksizler:	26 Beyaz		
	13 Gri		
	0 Siyah		

Ekranda kontrast oluşturmak için, her gruptan sadece bir renk kullanınız.

Programlarınızın CPC 464'ün monokrom ekranında iyi bir optik etki bırakması gerekiyorsa yanyana bulunan renklerin değerleri dörtten daha büyük olmalıdır. Renk değerleri büyüklüklerine göre renksiz monütörde artan bir parlaklık oluşturur ve iki komşu rengin renk değer farkı oldukça az ise, renkler seyreden kişi tarafından pek farkedilemez'er.

Bir örnek: Renkli monütörde açık mavi (2) ve kırmızı (3) renkleri birbirinden farklı gözükmesine rağmen, renksiz monütörde bu iki rengin birbirinden farklı olduğu pek anlaşılmaz.

Fakat mavi (1) ve açık kırmızı (6)'yı yada daha iyisi gök mavisi (11) ve kırmızı (3)'yı seçerseniz, renk değer farkı 5 ve 8 olduğu için renksiz monütörde bu iki rengin farklı olduğu açıkça anlaşılabilir.

2.4.3. Pencere

Metnin ekranda belirtilmesi için CPC 464'ün önemli bir ekran penceresi vardır.

Ekran penceresi, ekranın kullanıcı tarafından belirlenen ve aynı bağımsız bir ekran gibi çalışabilen bir alanıdır. Bu alanlarda programları listelemek ve çalıştırmak mümkündür.

CPC 464 üzerinde sekiz değişik ve birbirinden bağımsız pencere WINDOW komutu ile belirtilebilir. (Böyle bir pencere tekniği yalnız CPC 464'ün iki katı pahalı bilgisayarlarda mevcuttur!)

Birden çok ekran penceresi kullanırsanız, iki pencere tesadüfi yada belirli bir şekilde kesişirler. Bu durumda yeni penceredeki metin eski pencerede-

ki bilgileri siler.

Ayrıca ekran pencerelerinde grafik kullanırsanız, PLOT ve DRAW komutlarının ekran penceresi tariflerini bertaraf ettiğini ve enine bir alan çizdiğini bilmelisiniz.

Pozitif koordinat değerleri kullanırsanız, ORIGIN-komutu ile grafik penceresini tarif etmeniz mümkündür. Meselâ ORIGIN 320,200 komutu ekranın sağ üst köşesinde bir grafik penceresi oluşur.

2.4.4 Umlaute

Bilindiği gibi CPC 464 Alman Umlaut'larına sahip değildir. Bunu SYMBOL komutu ile ortadan kaldırabilir ve umlaut işaretlerini tarif edebilirsiniz.

Amaçlı bir şekilde, yazmaç tarafından aktif Alman cümlelerinde Umlaut olarak kabul edilip basılan ASCII değerleri kullanınız.

Aşağıdaki şekil her Umlaut için ASCII değerini göstermektedir. Yeni Umlaut'lar eski Umlaut'larla tuş takımı üzerinden çağrılabilmektedir:

Ä- 91-[	ä-123-SHIFT [
Ö- 92-\	ö-124-SHIFT @
Ü- 93-]	ü-125-SHIFT]
ß-126-CTRL 2	

Bu işaretleri tarif etmek için önce SYMBOL AFTER 91 komutu ile 91 değerinden sonraki ASCII işaretlerinin matrisleri değiştirilebilmeleri için RAM belleğe yüklenirler.

Aşağıdaki SYMBOL-komutları yeni işaretleri tarif eder:

```
SYMBOL  91,102,24,60,102,102,126,102,0
SYMBOL  92,102,60,102,102,102,102,60,0
SYMBOL  93,102,0,102,102,102,102,102,60,0
SYMBOY  123,102,0,120,12,124,204,118,0
SYMBOL  124,102,0,60,102,102,102,60,0
SYMBOL  125,102,0,102,102,102,102,62,0
SYMBOL  126,60,102,102,108,102,102,124,96
```

Özel işaretlerin matrisleri, yeni işaretlerin mevcut işaret cümlelerine uyacağı şekilde seçilir. Bu durumda Umlaut'ların değişken isimlerinde kullanılmayacağı belirtilmelidir, zira bunlar ASCII-değerlerinden dolayı BASIC-yorumlayıcı tarafından özel işaretler olarak işlenebilirler.

2.4.5 Örnekler

Bu bölümde öğrenilen konularda pratik uygulamalar yapılacaktır. Aşağıdaki program ekranın ortasına bir çember çizer:

```
10 REM *** Çember ***
20 MODE 2:CLEAR:DEG
30 ORIGIN 320,200:MOVE 320,399
40 FOR lauf=0 TO 360
50 DRAW 199*SIN(lauf),199*COS(lauf)
60 NEXT lauf
```

("lauf" burada değişken yerine kullanılmıştır)

Çarpma işlemini 50.satıra yükseltirseniz, çember ekranın kenarlarına değer; her iki değeri küçültürseniz, çember de küçülür. Şayet her iki çarpanı farklı değerde alırsanız, enteresan bir durum oluşur. Çarpanlar arasındaki fark ne kadar büyük olursa,

eğimi o kadar fazla olan bir elips oluşur.

Aşağıdaki program tesadüfi şekillenen, bölünen bir elips oluşturur:

```
10 REM *** Elipsler ***
20 MODE 2:CLEAR:DEFINT a-z:RANDOMIZE TIM
E:DEG
30 x=RND*640
40 y=RND*400
50 m=RND*50+50
60 n=RND*50+50
70 IF x+m>639 OR y+n>399 OR x-m<0 OR y-n
<0 THEN 30
80 ORIGIN x,y
90 MOVE 0,n
100 FOR lauf=1 TO 360
110 DRAW m*SIN(lauf),n*COS(lauf)
120 NEXT lauf
130 GOTO 30
```

30. ve 40. satırlarda Elips pozisyonunu belirleyen koordinat başlangıç noktası oluşur. 50. ve 60 satırlarda her iki çarpan 50'den 99'a kadar uzanan sayı alanından tesadüfi olarak seçilir.

Satır 70, Elipsin tamamen ekran alanı içersinde oluşup oluşmadığını kontrol eder, şayet durum doğru ise Elips belirtilir, aksi halde yeni rastgele sayılar verilir.

CPC 464'ün Grafiği matematiksel fonksiyonları ifade etmeyi kolaylaştırır. Aşağıdaki Program bir Sinüs eğrisi çizmektedir:

```
10 REM *** Sinüs Eğrisi ***
20 MODE 2:CLEAR:DEG
30 ORIGIN 0,200:DRAW 639,0
```

```
40 FOR lauf=0 TO 720
50 DRAW 640*lauf/720,199*SIN(lauf)
60 NEXT lauf
```

ORIGIN komutu koordinat başlangıç noktasının yerleştirilmesi için oldukça gereklidir, zira hiç bir grafik değeri ekranın görülebilen alanına ulaşması için sıralanmamalıdır. Sinüs salınımının ikinci yarısı negatif koordinatlar ile hatasız bir şekilde belirtilmektedir.

50. satırda değişken değeri iki tam sinüs salınımının ekranda oluşması için (720 = 2 x 360 derece) 720'ye bölünmüştür ve toplam ekran genişliğinden istifade etmek için 640 ile çarpılmıştır.

Bu satırlardan görüldüğü gibi problemin anlaşılmasında herhangi bir zorluk bulunmamaktadır.

Son program örneği ekran penceresi ile ilgilidir. Rastgele büyüklükteki tesadüfi bölünmüş ekran penceresi oluşur ve rastgele bir renk ile doldurulur:

```
10 REM *** Ekran Penceresi ***
20 MODE 0:DEFINT a-z:RANDOMIZE TIME
30 FOR pencere =0 TO 7
40 a=RND*20+1
50 b=RND*20+1
60 c=RND*25+1
70 d=RND*25+1
80 farbe=RND*26
90 farbregister=RND*15
100 BORDER farbe
110 WINDOW #pencere ,a,b,c,d
120 PAPER #Pencere ,farbregister
130 CLS #fenster
140 NEXT fenster
150 GOTO 30
```

40'dan 90'a kadar devam eden satırlarda rastgele değerler alınır ve 100 ile 130'uncu satırlarda bu değerler kaybedilir. Bu program kısmı bir FOR-NEXT döngüsü ile sona erer. Bu döngü bütün ekran penceresinin sıfırdan 7'ye kadar tam bir kez tanımlanmasını sağlar. Program döngüsünün tamamlanmasından, durum tekrar baştan başlar.

Bu program ile yeni ekran pencerelerinin eski pencere üzerine yazdığını açıkça görebilirsiniz.

2.5 Ses Jenaratörü

CPC 464'ün ses jeneratörü el kitabında birkaç örnekle açıklanmıştır. Ses jeneratörünün kullanılması için programlayıcının üç komutu vardır (SOUND, ENV ve ENT). Bu komutlar bilgisayarda mevcuttur.

Hesap yoluyla yaklaşık $2{,}5 \times 10^{84}$ ses ve gürültü zinciri bu komutlarla üretilebilir.

CPC 464'ün kullanım el kitabında ses jeneratörünün kullanılmasını sağlayan bu komutların anlamı ve uygulanması kısmen karmaşık olarak anlatılmıştır. Bundan dolayı bu bölümde ses jeneratörünün programlanma imkânları anlatılacaktır.

2.5.1 Sesler ve Gürültüler

CPC 464'ün ses jeneratöründe topları verebilen üç kanal ve gürültüleri verebilen bir kanal vardır.

Tonlar, SOUND komutunun frekans bölücü parametresinin sıfıra eşit olmayan bir değere yerleştirilmesi ile değişebilir. Gürültü SOUND-komutunun son parametresi olan "gürültü" ile kumanda edilir. Bu değer sıfıra eşit değilse, işitilebilen bütün frekansların karışmasından oluşan ve beyaz gürültü ola-

rak adlandırılan bir gürültü üretilir.

Şayet frekans bölücü ve gürültü parametrelerinin her ikisi de sıfırdan farklı bir değer alırlarsa, tonların ve gürültülerin aynı anda üretilmesi mümkün olur. Ton üretimi için üç kanal bulunduğu için CPC 464'e stereo bir tertibat bağlanması halinde stereo efektlerde elde edilebilinir, meselâ çift titreşimli bir melodide ilk titreşim sol hoparlörden, ikinci titreşim sağ hoparlörden verilebilir yada bunun tam tersi de olabilir.

Buna karşın gürültülerin verilmesi için yalnız bir kanal vardır ve gürültülerin stereo olarak verilmesi de mümkün değildir. Meselâ hoparlörün birinden tiz bir gürültü verilirken aynı zamanda diğer hoparlörden donuk bir gürültü verilemez. Gürültülerin sabit bir yönden yada her iki yönden de aynı anda verilmesi mümkündür.

2.5.2 SOUND

SOUND komutu ton verilmesi hususundaki en önemli kumanda işlemidir ve yedi parametreye kadar çalışabilir.

Aşağıdaki şekil bazı parametreleri sıraya göre belirtmiş ve onların değer alanını parantez içersinde göstermiştir:

Kanal durumu	(0'dan 255'e kadar)
Frekans bölümü	(0'dan 4095'e kadar)
Süre	(-32768'den 32767'ye kadar)
Ses şiddeti	(0'dan 15'e kadar)
Ses şiddeti kılıf eğri numarası	(0'dan 15'e kadar)
Frekans kılıf eğri numarası	(0'dan 15'e kadar)
Gürültüler	(0'dan 31'e kadar)

Şimdi bu parametreleri biraz açıklamaya çalışalım:

Kanal Durumu

Kanal durumu değer alanı tam bir Bayt'tır. Bu Bayt'ın her Bit'inin özel anlamı vardır:

0. Bit - SOUND sol kanal üzerine verme (1)
1. Bit - SOUND orta kanal üzerine verme (2)
2. Bit - SOUND sağ kanal üzerine verme (4)
3. Bit - Sol kanal ile senkronize etme (8)
4. Bit - Orta kanal ile senkronize etme (16)
5. Bit - Sağ kanal ile senkronize etme (32)
6. Bit - SOUND bekleme (64)
7. Bit - SOUND derhal uygulama (128)

Belirli bir fonksiyonu arzu ediyorsanız, kanal durumu değeri olarak parantez içindeki sayıyı kullanmalısınız. Daha fazla fonksiyonu uygulamak istediğinizde uygun Bit'lerdeki değerleri toplayıp, kanal durumu değeri olarak vermeniz gerekir, meselâ SOUND 135, SOUND'un üç kanal üzerinden verildiğini belirtir, yani 135 = 1 + 2 + 4 + 128.

Kanal durumunu 6. Bit ile verirseniz, bekleme durumu RELEASE komutu ile kaldırılınca, SOUND uygulama durumuna yani çalışma durumuna gelir.

Frekans Bölücü

Bir Tonun frekansı 125.000 frekans bölücü tarafından bölünerek hesaplanması sonucunda verilir. Yani frekans bölücü verilmesi gereken tonun yüksekliğini belirler.

Frekans bölücünün değeri 0 ile 4095 alanı

içersinde bulunduğundan dolayı 30,5 Hz'den 125 kHz'ye kadarki frekansların verilmesi mümkündür (frekans bölücü sıfıra yerleştirilirse, herhangi bir ton işitilmez).

Bu frekanslar size düşük geliyorsa aşağıdaki tabloya bakınız. Bu tablo size hangi yaşta hangi seslerin algılanabileceğini göstermektedir:

20 yaş - 16 Hz ile 20 kHz arasındaki sesler.
30 yaş - 16 Hz ile 16 kHz arasındaki sesler.
45 yaş - 16 Hz ile 12 kHz arasındaki sesler.
60 yaş - 16 Hz ile 6 kHz arasındaki sesler.

Gördüğünüz gibi yüksek tonların duyulması yaş durumunun artmasına bağımlı olarak düşmektedir. Müziğin en yüksek tonları bu değeri çok az ve çok seyrek bir şekilde aştığı için, 6 kHz'lık bu en yüksek işitme sınırı yeterlidir. Karakteristik tını titreşimleri artık algılanamayacağından, enstrumanlar arasındaki fark da zor algılanacaktır.

İnsanın dili çok değişik ve birbirini aşan frekansların bir karışımıdır, ve bu 1 kHz'den yüksek seslere ulaşamaz.

Akustik konusundaki bu küçük açıklama ile frekansların önemi daha iyi anlaşılmaktadır.

Süre

"Süre" parametresi tonun yada gürültünün uzunluğunu belirler. Değer pozitif ise 1/100 yani yüzde bir saniye uzunluğunda verilir, değer negatif ise, ses şiddeti eğrisi pozitif değere uygun olarak sık sık tekrarlanır.

Ses Şiddeti

"Ses şiddeti" parametresinin değeri ne kadar yüksekse, tonun ve gürültünün şiddeti de o kadar büyük olacaktır.

Ses şiddeti kılıf eğrisi olmaksızın 0 ile 7 arasındaki değer alanı 8 ile 15 arasındaki alana uygun düşer. Ses şiddeti kılıf eğrisi ile oldukça hassas şekilde ayırdedilir, zira ses şiddeti karışımı için 16 değer bulunmaktadır.

Ses Şiddeti Kılıf Eğri Numarası

Bu numara, SOUND komutu için hangi ENV durumunun geçerli olduğunu belirtir.

Frekans Kılıf Numarası

Bu numara, SOUND komutu için hangi ENT durumunun geçerli olduğunu belirtir.

Gürültü

SOUND komutunun son parametresi gürültünün yüksekliğini belirler. Değer ne kadar büyük olursa, gürültü o kadar donuk olur. Değer olarak sıfır· kaydedilirse, hiç bir gürültü işitilmez.

Parametrelerin bütünü için şunlar da söylenebilir: Bütün değerler tam sayı olmalıdır. Şayet tam sayı kaydetmezseniz, BASIC-Yorumlayıcı bu sayıyı en yakın tam sayıya yuvarlatır. Şayet değer sıfır olursa, herhangi bir şey belirtilmez. SOUND komutunun sonunda bulunursa, atılabilir, yada ihtiyaden arka arkaya iki virgül yerleştirilir.

2.5.3 ENV

ENV komutu ses şiddeti kılıf eğrisini belirler. 16 parametre kadar verilebilir. İlk parametre kılıf eğri numarasını belirler, diğer değerler üçlü gruplar halinde kılıf eğrisinin bazı kesimlerini teşkil ederler.

Aşağıda parametrelerin ve değer alanlarının bir düzenlemesi görülmektedir:

Kılıf eğri numarası (1'den 15'e kadar)
1.Adım sayısı (0'dan 127'ye kadar)
1.Adım büyüklüğü (-128'den 127'ye kadar)
1.Adım zamanı (0'dan 255'e kadar)

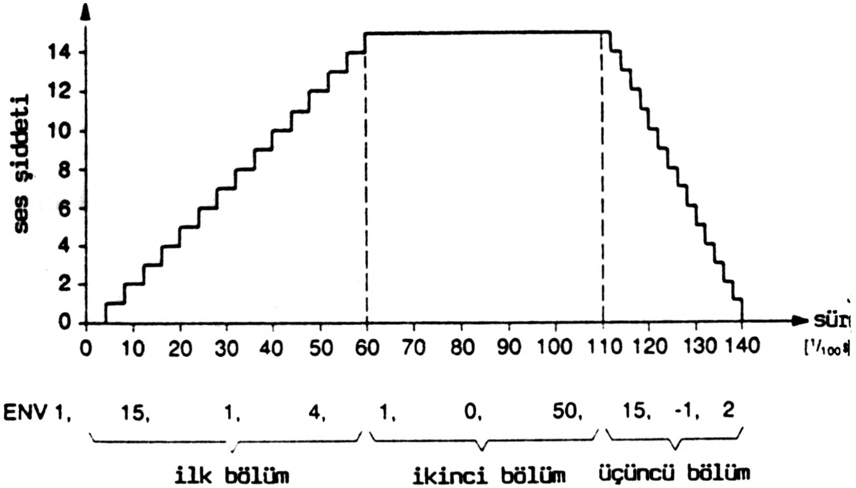
.
.
.
.
.
.

5.Adım sayısı (0'dan 127'ye kadar)
5.Adım büyüklüğü (-128'den 127'ye kadar)
5.Adım zamanı (0'dan 255'e kadar)

SOUND komutunun ses şiddeti büyük yada çok adımlarda 15 değerini kolayca aşabildiğinden dolayı, ses şiddeti = ses şiddeti MOD 16'dır, yani ses şiddeti 16 ses şiddeti sıfıra uygundur.

Bunun teorik bilgi kalmaması için, Resim 2.1, üç kılıf eğri kesiminden oluşan somut bir örnek göstermektedir.

İlk kılıf eğri kesimi 15 adımlık ses şiddetini 4 yüzde bir saniyede en yüksek değerini yükseltir, fakat bunun için ilgili SOUND komutunun ses şiddeti parametresi sıfıra eşit olmalıdır.



Resim 2.1. Ses şiddeti kılıf eğrisi örneği

(Değer sıfıra eşit değilse, ses şiddeti 15'i geçer ve tekrar sıfırdan başlar).

İkinci kılıf eğri kesimi yarım saniyelik süre için ses şiddetini dengede tutar ve üçüncü kesim 15. adımdaki ses şiddetini yüzde iki (2/100) saniyede çıkış değerine yani sıfıra düşürür.

ENV komutu ile tonun ses şiddetini periyodik olarak yükseltip alçaltan bir kılıf eğrisi teşkil ederseniz, bu komutla CPC 464 üzerinde mükemmel titreşim (ses) efektleri üretebilirsiniz.

2.5.4 ENT

ENT komutu frekans kılıf eğrisini belirler. 16 parametre kadar verilebilir. İlk parametre kılıf eğri

numarasını belirtir, diğer değerler üç grup halinde kılıf eğrisinin bazı kesimlerini teşkil ederler.

Aşağıda parametrelerin ve değer alanlarının bir düzenlemesini göreceksiniz:

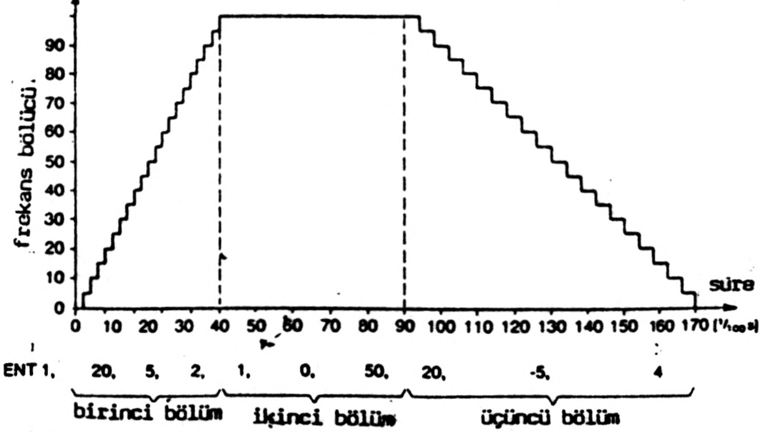
Kılıf eğri numarası (-15'den 15'e kadar)

1. Adım sayısı (0'dan 239'a kadar)
1. Adım büyüklüğü (-128'den 127'ye kadar)
1. Adım süresi (0'dan 255'e kadar)

.
.
.
.
.
.

5. Adım sayısı (0'dan 239'a kadar)
5. Adım büyüklüğü (-128'den 127'ye kadar)
5. Adım süresi (0'dan 255'e kadar)

Frekans kılıf eğrisinin kılıf eğri numarası ENT komutunda negatif bir ön işaret alabilir. Bu ön işaret, SOUND komutu tonunun sona erinceye kadar kılıf eğrisinin ne kadar sık tekrarlandığını gösterir. Kılıf eğri numarası negatif bir ön işaret almazsa, CPC 464 kılıf eğrisini sadece bir kez üretir.



Resim 2.2 Frekans kılıf eğrisi örneği

Pozitif adım genişliğinin tonu daha pes, negatif adım genişliğinin tonu daha tiz yaptığı bilinirse, ENT komutunun adım büyüklüğü boyutu oldukça basit anlaşılır.

Bu bilgilerin daha açık anlaşılması için üç kılıf eğri kesiminden oluşan somut bir örnek resim 2.2'de görülmektedir.

İlk kılıf eğri kesimi, ilgili SOUND komutunun frekans bölümünü 20. adıma yüzde iki saniyede yükseltir.

İkinci kılıf eğri kesimi, yarım saniyelik süre için frekans bölümünü dengede tutar, üçüncü kesim frekans bölümünü tekrar 20. adımdan yüzde dört saniyede başlangıç değerine düşürür.

Tonun frekansını periyodik olarak yükseltir ve alçaltırsanız, ENT komutu ile CPC 464 üzerinde siren efektleri elde edebilirsiniz.

2.5.5 Stereo

CPC 464'de sadece bir hoparlör vardır ve bundan dolayı tonlar mono olarak verilirler. Fakat bunun yanında stereo tonlar da üretilebilir. Stereo tertibatınız varsa bunu bilgisayarınıza bağlayabilir (bakınız konu 1.6.4) ve stereo tonları işitebilirsiniz.

Tonların ve gürültünün stereo olarak verilmesiyle (gürültülerde sınırlama vardır, bakınız konu 2.5.1) şaşırtıcı efektler elde edilebilinir. Çok tınılı şarkılar değişek hoparlörler üzerinden verilebilir, tonlar ve gürültüler akustik olarak hareketlendirilebilir.

Bir ton yada bir gürültüyü gerçeğe mümkün olduğu kadar uygun bir şekilde soldan sağa doğru gezdirmek için, değişik kanallardan verme sırasında şu sıra takip edilmelidir:

1. Kanal 1		
2. Kanal 1	Kanal 2	
3. Kanal 1		Kanal 3
4. Kanal 1	Kanal 2	Kanal 3
5. Kanal 1		Kanal 3
6.	Kanal 2	Kanal 3
7.		Kanal 3

Böylece aynı yükselen ve alçalan ton yada gürültü efektleri oluşacaktır.

Tabii ki sağdan sola doğru bir hareket de mümkündür. Yalnız düzenlemenin sırasını ters çevirmeniz gerekir.

2.5.6 Uygulamalar

Bu bölüm ses jenaratörü programlanması hususunda birkaç uygulama örneği sunmaktadır.

CPC 464 üzerinde belirli oktav ve nota değerlerinin müziksel tonlarını üretmek isterseniz, şu komut zincirini kullanacaksınız:

$$125000/(32{,}7032 * 2((ton - 1)/12 + oktav - 1))$$

Bu hesabın sonucu frekans bölücü olarak SOUND komutuna yerleştirilmelidir.

"Oktav" değişkeni için, Kontra-Oktavdan hareket ederek arzu edilen oktavın numarası verilmelidir. "ton" değişkeni için oktav dahilindeki çalınması gereken yarım tonun numarası verilir.

Tablo 2.1 CPC 464 üzerinde bu komut zinciri ile mümkün olan tüm oktav ve tonları göstermektedir.

Numara	Oktav	Ton
1	Kontra oktav	C
2	Büyük oktav	Cis
3	Küçük oktav	D
4	Tek vuruşlu oktav	Dis
5	Çift vuruşlu oktav	E
6	Üç vuruşlu oktav	F
7	Dört vuruşlu oktav	Fis
8	Beş vuruşlu oktav	G
9	Altı vuruşlu oktav	Gis
10	Yedi vuruşlu oktav	A
11	Sekiz vuruşlu oktav	Ais
12	Dokuz vuruşlu oktav	H

Tablo 2.1 CPC 464 üzerindeki oktavlar ve tonlar

Bildiğiniz gibi CPC 464'de ton verilmesi için üç değişik kanal vardır. Ton ve gürültü zincirlerinin SOUND komutları kanalların uygun döngü sırasına yüklenir ve mümkün olduğu kadar çabuk çalınır.

Tonları ve gürültüleri sadece bir kanal üzerine verirseniz, SOUND komutları ardı ardına çalınırlar.

Değişik kanalları kullanırsanız, SOUND komutları bu kanallar üzerine paralel bağlanırlar.

Bazen SOUND komutlarının ardı ardına değişik kanallara verilmesi arzu edilir. Bu durumda, diğer kanalların ton verme işlemi sona erdiyse birinci kanalın bekleme durumu yüklenir.

Aşağıdaki döngü, "kanal" üzerine kaydetme işlemi sona erinceye kadar devam eder:

```
WHILE SQ(kanal) > 127 : WEND
```

WHILE-WEND döngüsü her geçişte "kanal" döngü durumunun 7'nci Bit'inin yerleştirilip, yerleştirilmediğini, yani "kanal" kanalının aktif olup olmadığını kontrol eder. Durum bu şekilde ise döngü tekrar çalışır, aksi halde ise döngü sona erer ve aşağıdaki SOUND komutu bir sonraki kanalın döngü durumunu ton verme işlemi için yükleyebilir.

Aşağıda, bir roketin hareketi sırasındaki gürültü imkânlarından küçük bir bölümü görülmektedir.

```
10 REM *** Roketler Havalandı ***
20 FOR lauf=31 TO 1 STEP -1
30 SOUND 2,0,n,(lauf-2)/4,,,lauf
40 NEXT lauf
50 GOTO 10
```

Bu program örneğinin döngüsü tek kanatlı iki sineğe benzemektedir: Verilen gürültü, "lauf" değişkeninin yükseltilmesi sırasında, uzaklaşan raketin efekt'ini verebilmek için pes fakat aynı zamanda yüksek olur.

"n" değişkeni için 50 değerini yerleştirirseniz, büyük bir roketin hareket ettiğini, n = 5 değerinde ise küçük bir roketin hareket ettiğini anlarsınız.

Bu örnek programın gürültüsü orta kanal üzerinden verildiği ve böylece bazı "gürültüleri" bozmadığı için, WHILE-WEND döngüsünden vazgeçilebilinir.

CPC 464'ün stereo efekt'li ses ve gürültü olanakları hususundaki diğer örnekleri ve tafsilâtlı açıklamaları bölüm 3'de gösterge programında bulabilirsiniz.

2.6 Kesilmeler

Bölüm 1.2'den bildiğiniz gibi, CPC 464'ün mikro işleyicisinin üç değişik kesilme imkânı bulunmaktadır. Peki bu kesilme nedir? Doğru bir şekilde belirtilirse kesilme, mikro işleyicinin şu andaki işlevinin donanımsal yada yazılımsal olarak kesilmesidir. Genel olarak ifade edilirse kesilme, şu andaki program bölümünün işlevinin kesilmesi ve yeni bir program kesimine atlamasıdır.

Z-80-CPU'da üç mükemmel kesilme imkânı bulunmaktadır. CPC 464 Z-80-CPU ile bellenim sistemindeki gerçekleri gözönünde bulundurur. Basit BASIC komutlarıyla verimli kesilme işlemi imkânları elde edilebilir.

"ON SQ (kanal) GOSUB satır" komutu güzel bir örnektir. Bu komutu program bloğunun başına yerleştirirseniz, (kanal) Kanalının baş ton bekleme döngüsünde bu bloğun işlevi ani olarak kesilir ve "satır" satırından itibaren alt program çağrılır. ("kanal ve satır" birer değişken isimleridir). Alt programda bekleme döngü sırası tekrar doldurulur ve normal bir RETURN komutuyla program bloğunun terkedilen yerine geri atlanır.

Bu durum, CPC 464 kesilme anında otomatikman kesimin yapıldığı noktaya geri atladığı için mümkün olabilmektedir.

Diğer kesilme imkânları AFTER, EVERY ve REMAIN komutlarıyla yapılabilmektedir.

AFTER zaman, timer COSUB satır komutu "timer" zamanlayıcıyı "zaman" değeri ile yükler ve zamanlayıcıyı çalıştırır.

"zaman" süresi "timer" zamanlayıcı üzerine verilirse, "satır" program satırından itibaren alt prog-

ram çağrılmış olur.

"EVERY zaman, timer GOSUB satır" komutu "timer" zamanlayıcı "zaman" değeriyle yükler ve zamanlayıcıyı çalıştırır. "timer" zamanlayıcının süresinin dolduğu her defasında alt program "satır" satırından itibaren çağrılır ve aynı zamanda bu komutun yürütülmesi yeniden başlar.

Her iki komutta da alt program normal bir RETURN komutuyla sona erdirilir ve ana programın terkedilen yerine geri atlanır. "REMAIN timer" komutuyla "timer" zamanlayıcının geri kalan süresi soruşturulur ve aynı zamanda "timer" zamanlayıcı sıfıra geri yerleştirilir. Böylece zamanlayıcı basit olarak kesilebilir.

Daha fazla zamanlayıcının aynı anda bir program kesimini ve alt programı çağırması da mümkündür. Şayet her kesilme diğer bir alt programı çağırmak isterse, CPC 464 zorlanacaktır. Bundan dolayı zamanlayıcı belirli bir önceliğe göre düzenlenmiştir. Üçüncü zamanlayıcının yüksek öncelik hakkı vardır ve diğer zamanlayıcılardan önce çağrılması gereken alt programı belirler. Zamanlayıcı sıfır'ın en düşük önceliği vardır ve bütün diğer zamanlayıcılar ileridirler.

2.7 Çalışma Sistemi

Bir bilgisayarın çalışma sistemi cihazın, çevrenin kullanıcının kayıt yönetimleri ile ilgilidir.

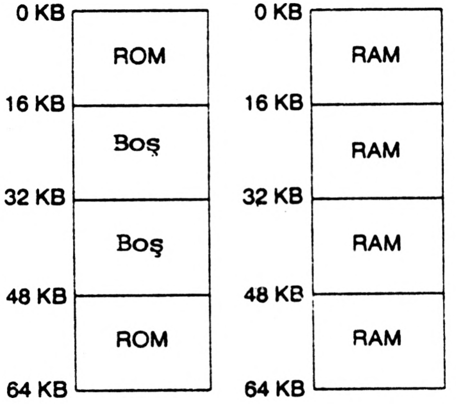
Aşağıdaki bölümde kullanıcı bilgilerinin özel yönetimi anlatılmıştır. Hangi bilgilerin CPC 464'ün çalışma belleğinin neresine yüklenebileceğini araştırınız. Bunun için verilen bilgilerin bellek örnekleri ve şekilleri açıklanmıştır.

Son bölüm, okunmayan programlardan kurtulmak

için pratik bir yöntemi açıklamaktadır.

2.7.1 Bellek Bölümleri

CPC 464'de 32 KBayt ROM ve 64 KBayt RAM bellek vardır. Bölüm 1'den de bildiğiniz gibi Z-80-CPU sadece 64 Kbayt bellek adresleyebilmektedir. CPC 464'ün konstrüktörleri RAM belleğe paralel olarak ROM belleği yerleştirerek bu problemi çözmüştür. ROM belleğin alt 16 KBayt'ı RAM belleğin alt 16 KBayt'ına paralel, ROM belleğin üst 16 KBayt'ı da RAM belleğin üst 16 KBayt'ına paraleldir. Resim 2.3 bu bölünmeyi açıklamaktadır.



Resim 2.3 CPC 464'deki ROM ve RAM Belleklerin Bölümleri

Z-80-CPU'nun kendi belleği ile karışmaması için, muhtelif RST-programları ROM ve RAM bunu kavrayabilmek için aktif yada pasif hale dönüşürler. Bu RST programları (yordamları), CPC 464'ün bellenim sistemine göz atmak için, Asamble uzmanları tarafından kullanılır. BASIC-programlayıcı, PEEK ve

POKE komutları ile CPC 464'ün ROM belleğini kavrayamadığından, bu imkânlara kapalı kalmaktadır.

RAM bellekte ise buna karşın istenildiği kadar PEEK ve POKE komutu kullanılabilinir. Tablo 2.2, RAM bellekte çalışma sisteminin hangi bilgilerinin nerede depolandığını açıkça ve teferruatlı olarak göstermektedir.

Adresler	İçeriği
0-63 arası	Kesilme ve bellek yönetimi için RST yordamı (Aynı isimli ROM-adreslerin kopya edilmesi)
64-321 arası	Uygulama vuruşu (üç sıfırla sona eren şu andaki komut yada program satırını kapsar.
322-367	Sıfır
368	BASIC-programının başlaması
368'den n'ye kadar	Üç sıfırla sona eren BASIC programı
12+1'de...	Değişken tablo başlangıcı
...43903'e kadar	Dizi belleği (ilk kaydedilen diziler başta bulunurlar)
43903	Dizi belleği sonu ve maximal HIMEN-değeri
43904-43911	240 ASCII işaretinin nokta matrisi
43912-43919	241 " " " "
43929-43927	242 " " " "
43928-43935	243 " " " "
43936-43943	244 " " " "
43944-43951	245 " " " "
43952-43959	246 " " " "
43960-43967	247 " " " "
43968-43975	248 " " " "
43976-43983	249 " " " "
43984-43991	250 " " " "
43992-43999	251 " " " "
44000-44007	252 " " " "
44008-44015	253 " " " "
44016-44023	254 " " " "
44024-44031	255 " " " "
44032-44195	Çalışma sistemi bilgileri
44196-44450	Kaydetme vuruşu (tek sıfırla sona eren şu anda kaydedilen ASCII-işaretlerini kap-

	sar.
44451-49151	Çalışma sistemi bilgileri
44592-44593	Puantör, BASIC-program başlangıcı -1'i gösterir.
44667-44668	Hali hazırdaki HIMEM-değeri
44669,44670	Maximal HIMEM-değeri
44673,44674	BASIC-program başlangıcı -1'i gösteren puantör
44675,44676	Değişken tablo başlangıcını gösteren puantör
44677,44678	Değişken tablo başlangıcını gösteren puantör
45197,45198	Dizi belleği başlangıcını gösteren puantör
45199,45200	Dizi belleği başlangıcını gösteren puantör
47360-47393	ROM-yordamları için giriş adresleri
47872-48441	ROM-yordamları için giriş adresleri
48589-48627	ROM-yordamları için giriş adresleri
49152-65535	Ekran belleği
49152-51151	Ekran bildirim pozisyonunun 1. satırı
51200-53199	" " " 2. "
53248-55247	" " " 3. "
55296-57295	" " " 4. "
57344-59343	" " " 5. "
59392-61391	" " " 6. "
61440-63439	" " " 7. "
63488-65487	" " " 8. "

Tablo 2.2 CPC 464'ün RAM-Bellek terminalleri

2.7.2 Bilgilerin Belleğe Kaydedilmesi

256'dan küçük sayıların kaydedilmesi zor değildir, çünkü bu bir POKE-komutuyla mümkündür.

Fakat bazen bir adresi RAM belleğe kaydetmek gerekli olabiliyor, meselâ göstergeyi değiştirmek için. Bir adres 65535'e kadar bir değer taşıya-

bildiği için, bu adres iki bellek satırına yerleştirilmelidir. Bir adresten iki POKE-değerine nasıl ulaşabiliriz?

İlk bellek satırına daima adresin LSB'si (en düşük Bayt'ı) POKE edilmeli ve ikinci bellek satırına da adresin MSB'si (en büyük Bayt'ı yüklenmelidir. Bunun için aşağıdaki hesaplamayı kullanınız:

MSB = INT (Adresse/256)
LSB = Adresse - MSB * 256

Bu operasyonun her iki sonucu uygun bellek yerlerine POKE edilmelidir-tamam!

2.7.3 ROM-Programları (Yordamları)

Bölüm 2.7.1'de gördüğünüz gibi, RAM bellekte ROM-programları için özel giriş adresleri vardır. Tablo 2.3, CPC 464'ün el kitabında uzman olmayan programlayıcılar için de, herhangi bir parametre verilmesi gerekmediği ve programların CALL komutu ile çağrılabildiği için, tamamen problemsiz bazı adresleri göstermektedir.

CPC 464'ün çalışma sistemi bu adresleri bizzat kendisi kullanmaz. Bu adreslerin herhangi birine ihtiyacınız yoksa, bu yerlere kendi özel RAM-programlarınızı sıralamak ve CALL komutu ile atlamak suretiyle, bu bellek alanından istifade edebilirsiniz.

2.7.4 Ekran Belleği

CPC 464'ün ekran belleği RAM belleğin üst 16 KByt'lık bölümünü kapsar. Aşağıdaki küçük program ekranı renk listesinden "bir" rengi ile doldurur.

Adresler	Fonksiyonlar
0	RESET
47878	Bir işaretin kaydedilmesini bekler (SHIFT yada CRTL değildir)
47896	Bir işaretin kaydedilmesini bekler (SHIFT yada CRTL değildir)
47941	BREAK komutunu otomatikman RESET komutu izler (CLEAR ve INPUT komutları bu programı kaldırırlar)
47944	BREAK komutunu önler (CLEAR ve INPUT komutları bu programı kaldırırlar)
47947	ESC tuşuna bir kez basma
47956	Ekranın boşaltılması (sadece çalışan programda kullanılabilir)
47959	Ekranın kapatılması (sadece çalışan programda kullanılır)
47980	BASIC-komutu CLS
48028	BASIC-komutu PRINT CHR$(24)
48091	BASIC-komutu CLG
48148	PAPER 0 : CLS komut zinciri
48238	Kaset-teyp motorunu çalıştırır.
48341	Kaset-teyp motorunu durdurur.
48307	Ton'un bırakılması
48310	Halihazırdaki tonu durdurur
48313	Halihazırdaki tonu çalıştırır
48409	Resmin tam olarak teşekkülünü bekler
48439	Giriş adres bloğu 47872-48441'i restore eder

Tablo 2.3 ROM-programları için giriş adresleri

```
10 FOR satır=1 TO 8
20 ilave=ilave +2048
30 FOR lauf=47104+ilave TO 49103+ilave
40 POKE lauf,255
50 next lauf
60 NEXT satır
```

("satır, ilâve, lauf" terimlerinin herbiri bir değişkendir)

Bu örnekten, ekran belleğine POKE edilmenin ne kadar karmaşık olduğunu görmektesiniz. CPC 464'ün kendilerine ASCII-işaretleri verilen, 2000 ekran pozisyonu vardır. Bilgisayarın işaretleri 8x8'lik nokta matrisinden oluştuğu için, her ekran veri pozisyonu sekiz satırdan ve her satırda yine sekiz noktadan oluşmuştur.

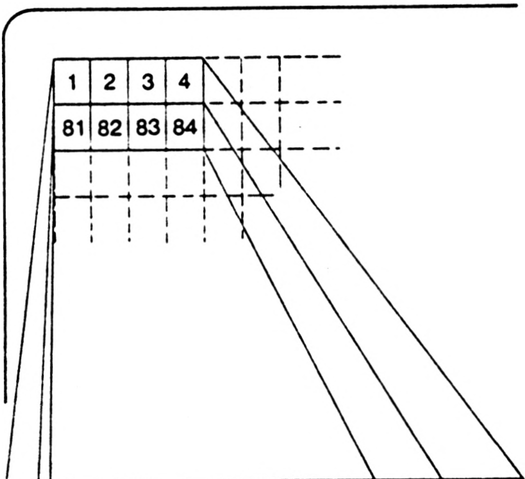
Ekran pozisyonunun satırları CPC 464'ün ekran belleğinde bir bayt (Byte) ile belirtilir. Bir ekran pozisyonunun satırları bellek içersinde birbiri arkasını takip ederek sıralanmazlar, aksine bütün ekran kayıt pozisyonunun bir satırı 2000 Bayt'lık blokta; ekran belleğinin ilk bloğu bütün ekran pozisyonlarının ilk satırını ve ikinci blokta bütün kayıt pozisyonlarının ikinci ekran satırını kapsayacak şekilde toplanır.

Şekil 2.4 ekranın sol üst köşesi için, ekran belleğinin hangi bellek satırlarının hangi ekran satırları ile ilgili olduğunu göstermektedir. Bu şema diğer ekran pozisyonları için uygun biçimde genişletilebilir.

CPC 464'deki her satır için bellek alanında tam bir Bayt'lık alan bulunduğu için, her ekran satırı için bilgilerin sadece belirli bir bölümü yüklenebilir. Bilgisayarın ekran modu, bir satırın kaç resim noktasının birbirinden bağımsız olarak birleşebileceğini

ve bu noktalar için kaç rengin mümkün olduğunu belirtir. Bir resim noktasının rengi yada resim noktaları grubunun rengi renk listesi ile tespit edilir. Bu listenin numaraları bir ikili sayının desimal (ondalık) değeri ile belirtilir ve bu ikili değerin rakamları uygun resim noktalarının bazı Bit'lerinden oluşur.

Bu sistem yüksek dağılan Mod 2'de daha basit gözlenebilir:

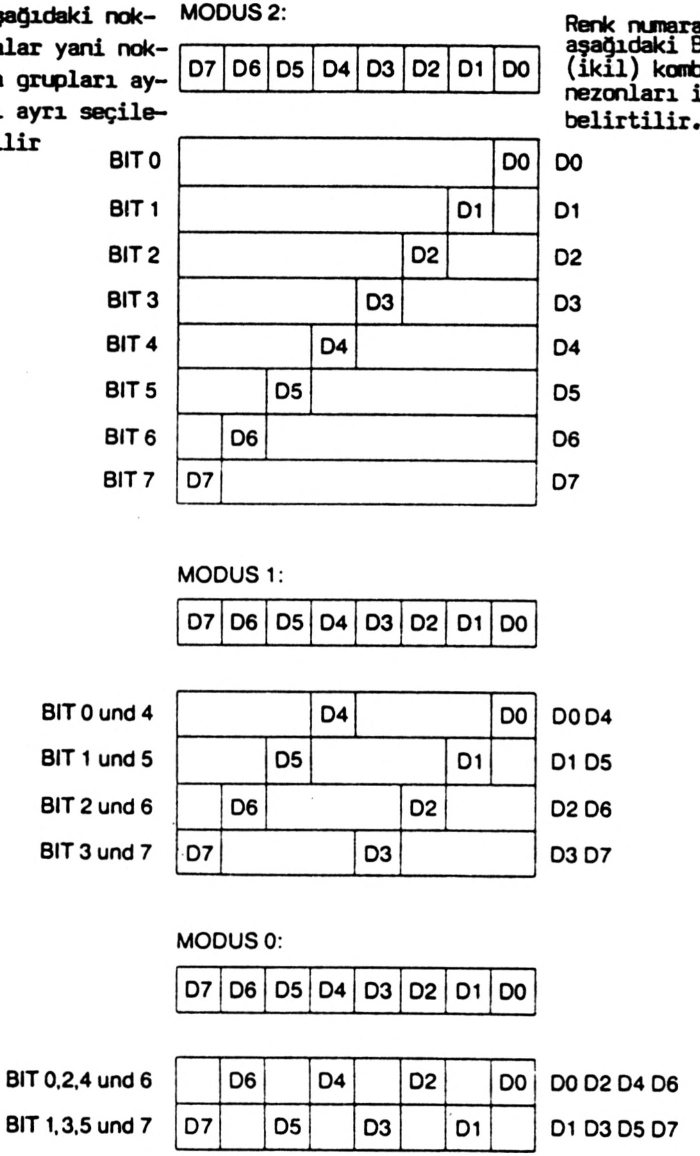


49152	49153	49154	49155
51200	51201	51202	51203
53248	53249	53250	53251
55296	55297	55298	55299
57344	57345	57346	57347
59392	59393	59394	59395
61440	61441	61442	61443
63488	63489	63490	63491
49232	49233	49234	49235
51280	51281	51282	51283
53328	53329	53330	53331
55376	55377	55378	55379
57424	57425	57426	57427
59472	59473	59474	59475
61520	61521	61522	61523
63568	63569	63570	63571

Resim 2.4 Ekran satırları için bellek satırları

Aşağıdaki noktalar yani nokta grupları ayrı ayrı seçilebilir

Renk numarası aşağıdaki Bit (ikil) kombinezonları ile belirtilir.



Resim 2.5 Ekran satır baytının bir bitinin fonksiyonu

Bir ekran satırının her resim noktası direkt olarak ilgili satır Bayt (çoklu)'ının bir Bit (ikil)'i ile belirtilir. Resim noktasının rengi uygun ikilinin (Bit'in) durumu ile tespit edilir. Şayet Bit bire yerleştirilmiş ise resim noktası renk listesinde "bir" rengini alacaktır. Bit sıfıra geri yerleştirilirse, resim noktası renk listesinden "sıfır" rengini alacaktır.

Diğer her iki Mod'da metod oldukça karmaşıktır:

Standart Mod 1 dört rengin ifadesini verebilmektedir, fakat iki sabit resim noktası aynı zamanda bir renk ile bağlanabilir. Bu her iki noktanın rengi renk listesi ile tespit edilmiştir. Bu renk listesinin numaraları iki resim noktası için her iki Bit'in binear (ikil) değerinden teşkil edilmiştir.

Çok renk modu "0" da dört resim noktasının rengi aynı zamanda 16 renk ile belirtilmelidir. Bu resim nokta gruplarının rengi, numaraları uygun Bit'lerden oluşan renk listesi ile sabittir.

Bu sistemin anlaşılmasını kolaylaştırmak için resim 2.5, ekran satır Bayt'ının hangi Bitlerinin hangi mod'da yerleştirilebileceğini ve resim listesi numaralarının bu Bit'lerden nasıl oluştuğunu göstermektedir.

Gördüğünüz gibi her Bit değişik mod'larda farklı fonksiyonlara sahiptir. Bu metod size bazı zorluklar çıkarıyorsa, bu hususta birkaç deneme yapmanız gerekecektir.

Ekran belleğine amaçsız bir şekilde POKE verilmesi anında CPC 464'de bir değişiklik olmaz, bunun için endişe etmeye gerek yoktur. Fakat ekran satır blokunu takip eden 48 boş Bayt'a değer kaydedilmemelidir. Bu durumda bilgiler ekran üzerinde kısmen belirirler ve orada herhangi bir şeyi değiştirirseniz, ekranda karmaşıklık ortaya çıkar.

2.7.5 BASIC-Program Satırlarının Şekli

Bilgisayarın çalışma belleğindeki program satırlarının ardarda sıralanması program metni olarak tanımlanır.

Program metnindeki bir BASIC-satırının şekli aşağıdaki gibi görülür:

1. Bayt	n satır uzunluğunun LSB'si
2. Bayt	n satır uzunluğunun MSB'si
3. Bayt	satır numarasının LSB'si
4. Bayt	satır numarasının MSB'si
5. Bayt	BASIC-kodları veya değişkenleri
6. Bayt	BASIC-kodları, değişkenleri, metin vs. (opsiyonel)
.	
.	
.	
.	
.	
.	
n-1. Bayt	BASIC-kodları, değişkenler, metin vs. (opsiyonel)
n. Bayt	satır sonu için 0 sayısı

Bu satır bir programın başında bulunuyorsa, bu satır 368. bellek satırında ilk Bayt'ta çalışmaya başlar.

Bu satırda programın sonu sözkonusu ise, n+1, n+2 ve n+3 bellek satırında program üç sıfırda sona erer.

Program satırın ortasındaki opsiyonel olarak işaretlenen baytlar, satır özel bir BASIC-kodundan daha fazla içerdiği zaman yalnızca bellekte görünürler.

2.7.6 BASIC-Kodları

Son bölümde BASIC-kodlarından bahsedilmektedir. Birçok okuyucu da BASIC-kodunun ne olduğu

sorusu oluşacaktır.

Şayet bir BASIC-komutu program metninde harf harf kaydedilirse, oldukça fazla yer kaplayacak, bu kaydedilen harfler değişken isimleriyle karışabilecek ve komutun yorumlayıcı tarafından algılanmasında büyük záman kaybı oluşacaktır. O halde bu yöntem birçok sebepten dolayı uygun değildir.

Çoğu bilgisayarlar her BASIC-komutuna, bir yada iki Bayt'tan oluşan sabit Kod numaraları düzenlemişlerdir.

Bu yöntem hem bellek alanından, hem de zamandan istifade etmektedir. Ayrıca komutların program metnindeki değişkenlerle karışmasını da önlemektedir.

CPC 464'ün BASIC-komut kod numaraları normal komutlarda 7 Bit'e yerleştirilmiş bir Bayt'tan, fonksiyon komutlarında ise 225 değerli bir Bayt'tan oluşmaktadır.

Tablo 2.4 CPC 464'ün bütün BASIC-komutlarının kod numaralarını göstermektedir.

BASIC-Token *	BASIC-Befehl	BASIC-Token	BASIC-Befehl	BASIC-Token	BASIC-Befehl
255+ 0	ABS	255+ 16	LOG10	255+ 66	HIMEM
255+ 1	ASC	255+ 17	LOWER$	255+ 67	INKEY$
255+ 2	ATN	255+ 18	PEEK	255+ 68	PI
255+ 3	CHR$	255+ 19	REMAIN	255+ 69	RND
255+ 4	CINT	255+ 20	SGN	255+ 70	TIME
255+ 5	COS	255+ 21	SIN	255+ 71	XPOS
255+ 6	CREAL	255+ 22	SPACE$	255+ 72	YPOS
255+ 7	EXP	255+ 23	SQ	255+113	BIN$
255+ 8	FIX	255+ 24	SQR	255+114	DEC$
255+ 9	FRE	255+ 25	STR$	255+115	HEX$
255+ 10	IMKEY	255+ 26	TAN	255+116	INSTR
255+ 11	INP	255+ 27	UNT	255+117	LEFT$
255+ 12	INT	255+ 28	UPPER$	255+118	MAX
255+ 13	JOY	255+ 29	VAL	255+119	MIN
255+ 14	LEN	255+ 64	EOF	255+120	POS
255+ 15	LOG	255+ 65	ERR	255+121	RIGHT$

BASIC-Token	BASIC-Befehl	BASIC-Token	BASIC-Befehl	BASIC-Token	BASIC-Befehl
255+122	ROUND	168	LOAD	213	WEND
255+123	STRINGS	169	LOCATE	214	While
255+124	TEST	170	MEMORY	215	WIDTH
255+125	TESTR	171	MERGE	216	WINDOW
255+127	VPOS	172	MIDS	217	WRITE
128	AFTER	173	MODE	218	ZONE
129	AUTO	174	MOVE		DI
130	BORDER	175	MOVER	220	EI
131	CALL	176	NEXT	227	ERL
132	CAT	177	NEW	228	FN
133	CHAIN	178	ON	229	SPC
134	CLEAR	179	ON BREAK	230	STEP
135	CLG	180	ON ERROR	231	SWAP
136	CLOSEIN		GOTO 0	235	TAB
137	CLOSEOUT	181	ON SQ	235	THEN
138	CLS	182	OPENIN	236	TO
139	CONT	183	OPENOUT	237	USING
140	DATA	184	ORIGIN	238	>
141	DEF	185	OUT	239	=
142	DEFINT	186	PAPER	240	> =
143	DEFREAL	187	PEN	241	<
144	DEFSTR	188	PLOT	242	< >
145	DEG	189	PLOTR	243	< =
146	DELETE	190	POKE	244	+
147	DIM	191	PRINT	245	−
148	DRAW	192	' (' = REM)	246	•
149	DRAWR	193	RAD	247	
150	EDIT	194	RANDOMIZE	248	.
151	ELSE	195	READ	249	
152	END	196	RELEASE	250	AND
153	ENT	197	REM	251	MOD
154	ENV	198	RENUM	252	OR
155	ERASE	199	RESTORE	253	XOR
156	ERROR	200	RESUME	254	NOT
157	EVERY	201	RETURN	255 + *n*	siehe oben!
158	FOR	202	RUN		
159	FOR	203	SAVE		
159	GOSUB	204	SOUND		
160	GOTO	205	SPEED		
161	IF	206	STOP		
162	INK	207	SYMBOL		
163	INPUT	208	TAG		
164	KEY	209	TAGOFF		
165	LET	210	TROFF		
166	LINE	211	TRON		
167	LIST	212	WAIT		

* (Token:Kodları Befehl:Komutları anlamına gelmektedir.)

Tablo 2.4 BASIC komutlarının kod numaraları

2.7.7 BASIC-Değişkenlerinin Şekli

Değişkenler program metninde görüldüklerinde, BASIC-kodları veya diğer bir metinle karışmaması için, ayrı şekilleri vardır.

Değişkenlerde üç farklı tip olduğundan dolayı, birbirlerinden çok az farklı üç şekil mevcuttur.

Reel değişkenler şu şekilde yüklenirler:

13, değişken ismin uzunluğu + 4,0, harf...harf +128

İnteger değişkenler aşağıdaki gibi görünürler:

2, değişken ismin uzunluğu +4,0, harf...harf + 128

Dizi değişkenlerinin şekli ise şöyledir:

3, değişken ismin uzunluğu + 4,0, harf...harf + 128

Gördüğünüz gibi, her üç şeklin ilk Bayt'ı değişkenin tipini belirtmekte ve değişken ismin son harfi 7 Bit'e yerleştirmekle yüklenir (ASCII-değeri 128'e yükseltilir).

Şayet değişkenlere değerleri programın çalıştığı anda verilirse, bu değerler program metnine yüklenemezler, aksine özel bir değişken tablosuna yerleştirilmelidirler.

Tablo 2.5 değişken tablosundaki değişik değişken tiplerinin şekillerini göstermektedir.

Bu dizi değişkeni metni değişken tablosunda değil, bilâkis dizi belleğinde gösterilmektedir. Değişken tablosundaki adres, değişken metnin dizi belleğinde hangi bellek satırından itibaren başladığını belirtmektedir.

Reel değişkenler	Integer değişkenler	Dizi değişkenler
0	0	0
0	0	0
Değişken ismin ilk büyük harfi	Değişken ismin ilk büyük harfi	Değişken ismin ilk büyük harfi
:	:	:
:	:	:
Değişken ismin son büyük harfi + 128	Değişken ismin son büyük harfi + 128	Değişken ismin son büyük harfi +128
4	1	2
Mantis	Sayı değerinin LSB'si	Metin uzunluğu
Mantis	Sayı değerinin MSB'si	Adresin LSB'si
Mantis	0	Adresin MSB'si
Mantis	-	0
Üstel fonksiyon	-	-
0	-	-

Tablo 2.5 Değişken tablosundaki değişken tiplerinin şekilleri

2.7.8 Okunamayan Programların Okunması

CPC 464 genel BASIC-komutu VERIFY'ye sahip olmadığından ve böylece SAVE edilen programın denenmesi mümkün olmadığından, bilgisayarınız bir programı okumayabilir.

Şayet bir emniyet kopyası hazırlamışsanız bu o kadar önemli olmayacaktır. Okunamayan programlarda bir kopyanın bulunması sözkonusu olduğundan, bu programa yatırdığınız iş yada para ekseriyetle kaybolacaktır.

Okuma hatasında CPC 464'ün bir suçu yoktur, aksine müzik kaydederken düşmeyen kasetin (dropout) hatalı bant yeri bilgi yükleme ve kaydetme esnasında önemli bilgi boşluklarına sebep olmaktadır.

(Programın yeniden yüklenmesi sırasında aynı yerde bir okuma hatası oluşursa, drop-out yani bant'taki hatalı yeri anlayabilirsiniz. Daha fazla drop-out'lu kasetleri sadece müzik kaydı için kullanınız!)

Bilgisayarınız artık program yüklemek istemezse ve bilgiler kaybolursa ne yapabilirsiniz?

Bunun için de, bütün programın bir bölümünü kurtarabileceğiniz bir metod vardır: Bilgisayarınızı kapatınız ve tekrar açınız. Okunamayan programı yükleyiniz. Şayet "Load error b" oluştuysa şansınız var demektir, CPC 464 okuma hatası bulunmasına rağmen komple programı çalışma belleğine yükler. "Loard error a" durumunda ise bilgisayar programı sadece hatalı yere kadar yükler, devamını yüklemez. Birçok denemeden sonra "Load error b" durumu yakalanabilir.

Bu durumda bellekte bütün program yada programın bir bölümü bulunur ama onu listeleyemezsiniz. Bu durum CPC 464'ün BASIC-Program alanının ilk iki bellek satırını (368 ve 369) sıfıra yerleştirdiğinden kaynaklanmaktadır. BASIC-program alanının bu her iki yeri BASIC-programının ilk satırının satır uzunluğunu gösterir. Bu satırlar sıfırda bulunuyorsa programın ilk satırının uzunluğunun sıfır olduğu ve bunu müteakip satırların mevcut olduğu anlaşılır. BASIC-yorumlayıcı UST-komutunun verilmesinde bu işareti görür ve programın bittiğini sanarak son efektte hiç bir şey listelemez.

Programı tekrar listelenir hale getirmek için, her iki bellek satırına programın başlangıç değerlerini yazmanız gerekir. Bölüm 2.7.5'den de bildiğiniz gibi 368. bellek satırı satır uzunluğunun alt Bayt'ını, bellek satırı 369 satır uzunluğunun üst Bayt'ını kapsar (genel olarak "sıfır").

Gördüğünüz gibi 368'e satır uzunluğunun LSB'si 369 satır uzunluğunun MSB'si yüklenmelidir. Programın ilk satırı 255 Bayt'tan daha uzun olduğu için 369'uncu bellek satır değişmemiş olarak kalabilir.

İlk program satırının satır uzunluğunu basit bir deneme ile bulabilirsiniz, bu en azından 6 Bayt uzunluğunda olmalıdır (bakınız konu 2.7.5) ve değeri de başka bir tahminle belirlenir.

Tahmini değeri 368'inci bellek satırına POKE ediniz. Programı doğru olarak listelerseniz, isabet etmiş olursunuz, aksi halde bunu tekrar denemelisisiniz.

İlk program satırının sonu için PEEK komutu ile sıfırı ararsanız diğer bir alternatif ortaya çıkar. Sıfırı bulduğunuzda (Program metnindeki değişken şeklinin de sıfır değeri olduğuna dikkat ediniz) sıfır bellek satır adresinden 367'yi çıkarır ve elde edilen değeri 368'inci satıra yüklersiniz.

Bütün hepsi POKE'lenince program listelenmelidir, aksi halde hata yapar ve hepşeye baştan başlarsınız.

Şimdi programı çalıştırmayı deneyiniz! Değişken tablo göstergesi 370'inci bellek satırını gösterir ve programı çalıştırırsanız verilen değişken değerleri BASIC-programının başına yazabilirsiniz. Böylece er yada geç tamamlayabilirsiniz.

Programı çalıştırmadan yada edite etmeden önce değişken tablo göstergesini program sonuna yerleştirmeniz gereklidir. Program sonunu arka arkaya üç sıfırı aramak suretiyle bulabilirsiniz. Şimdi 370'inci bellek satırındaki muhtevayı 378'inci satıra kadar not ediniz (şayet i değişkeni tarafından üzerine yazılırsa) ve Direkt mod'da aşağıdakileri kaydediniz.

```
FOR i=379 TO 43901:IF PEEK(i)=0 AND PEEK(i+1)=0
AND PEEK(i+2)=0 THEN i=i+3:PRINT i ELSE NEXT i
```

Bir müddet sonra (birkaç dakika sürebilir, sabırsızlanmayın) bu komut zinciri tarafından bir sayı verilir. Bu değer değişken tablosunun yeni başlangıcını belirtir.

```
POKE 4476,i-INT(i/256)*256
POKE 44676,INT(i/256)
POKE 44677,i-INT(i/256)*256
POKE 44678,INT(i/256)
```

Yukardaki komutlarla değişken tablosu göstergesini düzeltebilirsiniz.

Böylece 370'den 378'e kadarki bellek satırlarını not edilen değerlerle düzeltebilirsiniz, çünkü bu bellek alanı i değişkeninin ara yüklenmesi için kullanılmıştır.

Şimdi programınız tekrar çalışabilir ve edite edebilir.

"Load error a" durumu ortaya çıktığında, CPC 464 ilk okuma hatası oluştuğunda programın okunmasını keser ve siz programın geri kalanını mecburen yeniden düzenlersiniz.

Buna karşın "Load error b" durumu oluştuğunda, bilgisayar programı tamamen yükler ve okuma hatası herhangi bir program satırında bulunur.

Yükleme esnasında hatanın zamansal olarak ortaya çıkmasıyla hatanın nerede olduğunu yaklaşık olarak kestirebilirsiniz. Ya programı baştan aşağı tararsınız ve uygun satırları düzeltirsiniz yada programın çalışması esnasında bir hata ikazının oluşmasını beklersiniz. Okuma hatasının tasfiyesi yeterliyse

meselâ değişken ismin verilişi değişmişse, bu şartlar altında gerçekten uzun yada belki boşuboşuna bir hata ikazı bekleyebilirsiniz.

Bu yöntem okunamayan programlardan kurtulmak için size basit gelmiyorsa, CPC 464'deki bellenim süreçlerinin anlaşılması size bunu kolaylaştırır ve size zaman kazandırabilir.

3

CPC 464'ÜN YAZILIM SİSTEMİ

Bu bölüm oldukça faydalı ve uygulanmış programlardan ve oyunların programlama yönergelerinden oluşmaktadır.

Programların sayfalar süren listelerinden korkmayınız, zira CPC 464'ün marifetleri ve hünerleri sadece somut örneklerde daha iyi görülmektedir. Bu bölümün yazılım sistemini baştan aşağı tetkik ederseniz, her programın sayısız yordamlarını görebilirsiniz. Bu yordamlar size kendi programlarınızı hazırlamada yardımcı olacaklardır.

Bütün programlar kendi kullanım imkânlarını açıklamaktadırlar. Bundan dolayı daha fazla açıklamaya gerek duyulmamıştır. Bir programın özel yordamlarının anlaşılması için uygulanması zor olan bütün programların fonksiyonlarının belirlenmesi gerekir.

Bu bölümün yazılım sisteminde gerçekten fazla metin verildiği için bütün programlarda Mod 2 kullanılır. Aynı zamanda bütün programlar Alman Umlaut'larını kullanırlar. Bu Umlaut'ların CPC 464'e ön yükleme yapılması ve bu bölümün yazılım sistemine göz atmak için ilk konunun programını kullanınız.

3.1 İçindekiler

```
100 REM   *** İÇİNDEKİLER ***
110 MODE 2:WIDTH 80:MEMORY 43903
120 KEY 128,"LIST ":KEY 138,"EDIT ":KEY
139,"RUN"+CHR$(13):KEY 140,"PRINT#n,"+CH
R$(34)
130 SYMBOL AFTER 91
140 SYMBOL 91,102,24,60,102,102,126,102,
0
150 SYMBOL 92,102,60,102,102,102,102,60,
0
160 SYMBOL 93,102,0,102,102,102,102,60,0
170 SYMBOL 123,102,0,120,12,124,204,118,
0
180 SYMBOL 124,102,0,60,102,102,102,60,0
190 SYMBOL 125,102,0,102,102,102,102,62,
0
200 SYMBOL 126,60,102,102,108,102,102,12
4,96
210 PRINT STRING$(80,"_")
220 PRINT TAB(15);" *** İÇİND
EKİLER ***
230 PRINT STRING$(80,"_"):PRINT
240 PRINT"          GOSTERI          ;TAB
(50);"GOST";TAB(70);"()":PRINT
250 PRINT"          Aylık Takvim          ";T
AB(50);"MONAT";TAB(70);"()":PRINT
260 PRINT"BIORITM";TAB(50)
;"BIO";TAB(70);"()":PRINT
270 PRINT"FINANS HESAPL
ARI  ";TAB(50);"FINANSLAR";TAB(70);"()":P
RINT
280 PRINT"      FONKSIYON GRAFIĞI       ";T
AB(50);"FONKSIYON";TAB(70);"()":PRINT
290 PRINT"Soruşturma programı"
;TAB(50);"SORUŞTURMA";TAB(70);"()":PRINT
```

```
300 PRINT "ADRESLERIN YÖNETIMI e r w a l t
'I     ";TAB(50);"ADRESLER ";TAB(70);"()":P
RINT:PRINT
310 PRINT CHR$(7);:PRINT"Kaseti arzu edilen
programın band numarasına kadar sarınız
ve uygun programı büyük harflerle ve kısaltıl
mış olarak yükleyin"

320 END
```

100 Programın adı
110 Ön yükleme
120 İşlev tuşlarının yerleştirilmesi
130 ASCII'den 91 değerinin matrisinin RAM belleğe kopya edilmesi
140 işaretinin ifade edilmesi
150 Ö işaretinin ifade edilmesi
160 Ü işaretinin ifade edilmesi
170 işaretinin ifade edilmesi
180 ö işaretinin ifade edilmesi
190 ü işaretinin ifade edilmesi
200 işaretinin ifade edilmesi

210 Program adının
230 Kaydedilmesi

240'dan
300'e kadar program adının ve kısalmaların, band numaralarının kaydedilmesi

310 Metnin kaydedilmesi
320 Programın sonu

3.2 Gösteri Programı

```
100 REM *** GÖSTERİ PROGRAMI ***
110 MODE 2:CLEAR:DEFINT a-z:RANDOMIZE TI
ME:DEG
120 LOCATE 30,13:PRINT"CHAOS IM COMPUTER
"
200 REM
210 REM *** Özel Atışlar ***
220 ENV 1,15,-1,1
230 SOUND 7,0,-5,15,1,0,15
300 REM
310 REM *** Makineli Tüfek 1 ***
320 SOUND 1,4095,200,15,0,0,10
330 WHILE SQ(1)>127:WEND
400 REM
410 REM *** Makineli Tüfek 2 ***
420 ENV 1,5,-1,1
430 FOR lauf=1 TO 25
440 SOUND 4,0,10,15,1,0,10
450 WHILE SQ(4)>127:WEND
460 NEXT lauf
500 REM
510 REM *** Laser Ateşi ***
520 FOR laser=1 TO 5
530 FOR lauf=31 TO 1 STEP -1
540 SOUND 1,0,1,15,0,0,lauf
550 NEXT lauf
560 FOR lauf=1 TO 31
570 SOUND 4,0,1,15,0,0,lauf
580 NEXT lauf
590 NEXT laser
600 REM
610 REM *** patlama ***
620 FOR lauf=1 TO 31
630 SOUND 4,0,10,15,0,0,lauf
640 NEXT lauf
```

```
700 REM
710 REM *** Siren  ***
720 ENV 1,5,-1,20,5,1,20
730 ENT -1,100,1,1,100,-1,1
740 SOUND 2,60,-3,15,0,1
750 WHILE SQ(4)>127:WEND
800 REM
810 REM *** Marten Kornası ***
820 ENV 1,15,1,1,1,0,40,5,-1,1
830 FOR lauf=1 TO 5
840 SOUND 1,190,60,0,1
850 WHILE SQ(1)>127:WEND
860 SOUND 4,142,60,0,1
870 WHILE SQ(4)>127:WEND
880 NEXT lauf
900 REM
910 REM *** roketler start ***
920 FOR lauf=31 TO 4 STEP -1
930 SOUND 7,0,50,(lauf-2)/4,0,0,lauf
940 NEXT lauf
1000 REM
1010 REM *** Helikopter ***
1020 LOCATE 30,13:PRINT"HELIKOPTERDE
UÇUŞ"
1030 ENV 1,15,1,1
1040 FOR lauf=30 TO 15 STEP -1
1050 SOUND 1,0,lauf,0,1,0,lauf
1060 WHILE SQ(1)>127:WEND
1070 NEXT lauf
1080 SOUND 5,0,-30,0,1,0,15
1090 WHILE SQ(1)>127:WEND
1100 FOR lauf=15 TO 30
1110 SOUND 4,0,lauf,0,1,0,lauf
1120 WHILE SQ(4)>127:WEND
1130 NEXT lauf
1140 ENV 1,15,1,5,15,-1,10
1150 SOUND 4,0,250,0,1,0,31
```

```
2000 REM
2010 REM *** Demiryolu ***
2020 MODE 1:LOCATE 15,13:PRINT"DEMİRYOLU"
"
2030 kanal=1
2040 FOR lauf=24 TO 6 STEP -2
2050 GOSUB 10000
2060 NEXT lauf
2070 kanal=3
2080 GOSUB 20000
2090 ENV 1,15,1,1,15,-1,2
2100 SOUND 7,0,-20,0,1,0,3
2110 WHILE SQ(2)>127:WEND
2120 kanal=6
2130 GOSUB 20000
2140 kanal=4
2150 FOR lauf=6 TO 16 STEP 2
2160 GOSUB 10000
2170 NEXT lauf
3000 REM
3010 REM *** Örnek 1 ***
3020 MODE 2
3030 FOR y=0 TO 399 STEP 5
3040 MOVE 0,y:DRAW 639,399-y
3050 NEXT y
3060 FOR x=0 TO 639 STEP 5
3070 MOVE x,0:DRAW 639-x,399
3080 NEXT x
3090 FOR warten=1 TO 2000:NEXT warten
3100 REM
3110 REM *** Örnek 2 ***
3120 CLS:MOVE 0,0
3130 FOR lauf=0 TO 399 STEP 6
3140 DRAW 639,lauf:DRAW 639-lauf,399:DRA
W 0,399-lauf:DRAW lauf,0
3150 NEXT lauf
3160 FOR warten=1 TO 2000:NEXT warten
```

```
3200 REM
3210 REM *** Örnek 3 ***
3220 CLS:MOVE 0,0
3230 FOR x=0 TO 639 STEP 8
3240 MOVE 0,0:DRAW x,399
3250 MOVE 639,399:DRAW x,0
3260 MOVE 639,0:DRAW x,399
3270 MOVE 0,399:DRAW x,0
3280 NEXT x
3290 FOR warten=1 TO 2000:NEXT warten
3300 REM
3310 REM *** Sinüs Eğirisi ***
3320 CLS:ORIGIN 0,200:DRAW 639,0
3330 FOR lauf=0 TO 720
3340 DRAW 640*lauf/720,199*SIN(lauf)
3350 NEXT lauf
3360 FOR warten=1 TO 2000:NEXT warten
3400 REM
3410 REM *** daire ***
3420 CLS:ORIGIN 320,200:MOVE 320,399
3430 LOCATE 38,13:PRINT" Daire "
3440 FOR lauf=0 TO 360
3450 DRAW 199*SIN(lauf),199*COS(lauf)
3460 NEXT lauf
3470 FOR lauf=0 TO 360
3480 MOVE 0,0
3490 DRAW 199*SIN(lauf),199*COS(lauf)
3500 NEXT lauf
3510 FOR warten=1 TO 2000:NEXT warten
4000 REM
4010 REM *** hatlar ***
4020 CLS:ORIGIN 320,200
4030 x=INT(RND*5+2)
4040 y=INT(RND*5+2)
4050 FOR lauf=1 TO 3600 STEP 6
4060 z=100*COS(lauf)
4070 MOVE z,z
```

```
4080 DRAW 199*COS(lauf/x),199*SIN(lauf/y
)
4090 NEXT lauf
4100 GOTO 4020
10000 REM
10010 REM *** hızlandırma ve yavaşlatma alt
sistemi ***
10020 ENV 1,15,1,lauf/2,15,-1,lauf/2
10030 SOUND kanal,0,-1,0,1,0,lauf
10040 WHILE SQ(kanal)>127:WEND
10050 RETURN
20000 REM
20010 REM *** sabit hız alt sist
emi ***
20020 ENV 1,15,1,2,15,-1,2
20030 SOUND kanal,0,-5,0,1,0,4
20040 WHILE SQ(2)>127:WEND
20050 RETURN
```

Dikkat : Warten bir değişken değerdir.

100 Program ismi
110 Ön yükleme
120 Metin basımı

200 Atışların kaydedilmesi için yordam
220 Ses şiddeti eğrisinin tarif edilmesi
230 Eğrinin üç kanal üzerine beş kez verilmesi

300 Makinalı tüfek 1'in verilmesi için yordam
320 Tiz sesin paralel verilmesi ve gürültünün iki saniye için sol kanala verilmesi
330 Döngü gürültünün sona ermesini bekliyor

400 Makinalı tüfek 2'nin verilmesi için program
420 Ses şiddeti eğrisinin tarif edilmesi
430 Eğrinin yirmibeş kez kaydedilmesi için döngü başlangıcı
440 Gürültünün ses şiddeti eğrisiyle verilmesi

450 Döngü gürültü verilişinin sona ermesini bekliyor
460 Eğrinin yirmibeş kez kaydedilmesi için döngü sonu

500 Laser ateşinin verilmesi için program
520 Beş kez laser atışı için döngü başlangıcı
530 Gürültünün geri sayması için döngü başlangıcı
540 Gürültünün sol kanal üzerine verilmesi
550 Gürültünün geri sayması için döngü sonu
560 Gürültünün ileri sayması için döngü başlangıcı
570 Gürültünün sağ kanal üzerine verilmesi
580 Gürültünün ileri sayması için döngü sonu
590 Beş kez laser atışı için döngü sonu
600 Patlamanın kaydedilmesi için program
620 Gürültünün ileri sayması için döngü başlangıcı
630 Gürültünün sağ kanal üzerine verilmesi
640 Gürültünün ileri sayması için döngü sonu

700 Sirenin kaydedilmesi için program
720 Alçalan ve yükselen siren sesi için ses şiddeti eğrisinin tarif edilmesi
730 Sesin uğultusu için frekans eğrisinin tarif edilmesi
740 Siren sesinin orta kanal üzerine üç kez verilmesi
750 Döngü sağ kanal üzerindeki gürültünün (patlama) sona ermesini bekliyor

800 Martin kornasının kaydedilmesi için program
820 Yumuşak ses için ses şiddeti eğrisinin tarif edilmesi
830 Martin kornasının beş kez verilmesi için döngü başlangıcı
840 Tiz sesin sol kanal üzerine verilmesi
850 Döngü sesin sona ermesini bekliyor
860 Bas sesin sağ kanal üzerine verilmesi
870 Döngü sesin sona ermesini bekliyor
880 Martin kornasının beş kez verilmesi için döngü sonu

900 Roketlerin havalanması için program
920 Gürültünün geri sayması ve ses şiddetinin azaltılması için döngü başlangıcı
930 Gürültünün her üç kanala verilmesi
940 Gürültünün geri sayması ve ses şiddetinin azaltılması için döngü sonu

1000 Helikopter uçuşunun kaydedilmesi için program

1020 Metin basımı
1030 Ses şiddeti eğrisinin tarif edilmesi
1040 Gürültünün geri sayması ve kısaltılması için döngü başlangıcı
1050 Gürültünün sol kanala verilmesi
1060 Döngü gürültünün sona ermesini bekliyor
1070 Gürültünün geri sayması ve kısaltılması için döngü sonu
1080 Gürültünün sol ve sağ kanala otuz kez verilmesi
1090 Döngü gürültünün sona ermesini bekliyor
1100 Gürültünün ileri sayması ve uzatılması için döngü başlangıcı
1110 Gürültünün sağ kanala verilmesi
1120 Döngü gürültünün sona ermesini bekliyor
1130 Gürültünün ileri sayması ve uzatılması için döngü sonu
1140 Durdurma için ses şiddeti eğrisi tarif edilmiştir.
1150 Gürültünün sağ kanala verilmesi

2000 Trenin hareketi için program
2020 Metin basımı
2030 "sol" kanalın belirtilmesi
2040 Gürültünün geri sayması ve kısaltılması için döngü başlangıcı
2050 Hızlandırma alt programının çağrılması
2060 Gürültünün geri sayması ve kısaltılması için döngü sonu
2070 "sol orta" kanalın belirtilmesi
2080 "sabit hız" alt programının çağrılması
2090 Ses şiddeti eğrisinin tarif edilmesi
2100 Gürültünün üç kanal üzerine yirmi kez verilmesi
2120 "sağ orta" kanalın belirtilmesi
2130 "sabit hız" alt programının çağrılması
2140 "sağ" kanalın belirtilmesi
2150 Gürültünün ileri sayması ve azaltılması için döngü başlangıcı
2160 Yavaşlatma alt programının çağrılması
2170 Gürültünün ileri sayması ve uzatılması için döngü sonu
3000 Örnek-1'in verilmesi için program
3020 Yüksek çözülen mod'da ekranın silinmesi
3030 y-Koordinatının yükseltilmesi için döngü başlangıcı
3040 Bir hatın ekran üzerine verilmesi

3050 y-Koordinatının yükseltilmesi için döngü sonu

3060 x-Koordinatının yükseltilmesi için döngü başlangıcı

3070 Bir hatın ekran üzerine verilmesi

3080 x-Koordinatının yükseltilmesi için döngü sonu

3090 Bekleme döngüsü

3100 Örnek-2'nin verilmesi için program

3120 Ekranın silinmesi ve grafik kursörünün pozisyon alması

3130 Kaydırma faktörü için döngü başlangıcı

3140 Kaydırılan dörtgenin verilmesi

3150 Kaydırma faktörü için döngü sonu

3160 Bekleme döngüsü

3200 Örnek-3'ün verilmesi için program

3220 Ekranın silinmesi ve grafik kursörünün pozisyon alması

3230 x-Koordinatının yükseltilmesi için döngü başlangıcı

3240 Çizgi (hat) sol alttan başlıyor

3250 Çizgi sağ üstten başlıyor

3260 Çizgi sağ alttan başlıyor

3270 Çizgi sol üstten başlıyor

3280 x-Koordinatının yükseltilmesi için döngü sonu

3290 Bekleme döngüsü

3300 Ekranın silinmesi, koordinat sıfır noktasının yerleştirilmesi ve x-Ekseninin verilmesi

3330 İki tam sinüs dalgası için döngü başlangıcı (720 derece)

3340 x-Koordinatının, iki dalganın (720 derece) tam bir ekran genişliği (640 nokta) kapsadığının hesaplanması y-Koordinatının, dikeyin ekran yüksekliğinin (200 nokta) tam yarısının hesaplanması

3350 İki tam sinüs dalgası için döngü sonu

3360 Bekleme döngüsü

3400 Çemberin verilmesi için program

3420 Ekranın silinmesi, koordinat sıfır noktasının yerleştirilmesi ve grafik kursörünün çember başlangıcında pozisyon alması

3430 Metin basımı

3440 Tam bir çember için döngü başlangıcı (360 derece)

3450 x ve y koordinatlarının, çemberin ekran çerçevesinin üst ve alt kenarına kadar uzandığının hesaplanması ve çemberin verilmesi

3460 Tam bir çember için döngü sonu

3470 Tam bir çember için döngü sonu (360 derece)

480 Grafik kursörünün çemberin orta noktasında bulunması
3490 Çemberin orta noktasından çemberin kenarına kadar uzanan çizginin verilmesi
3500 Tam bir çember için döngü sonu
3510 Bekleme döngüsü

4000 Çizgilerin verilmesi için program
4020 Ekranın silinmesi ve koordinat 0 noktasının yerleştirilmesi
4030 x koordinatı için 1 ile 7 arasındaki rastgele değerin bulunması
4040 y koordinatı için 1 ile 7 arasındaki rastgele değerin bulunması
4050 Sayılar ve çizgilerin pozisyonu için döngü başlangıcı
4060 Çizgi başlangıç koordinatlarının tarif edilmesi
4070 Grafik kursörünün çizgi başlangıç koordinatında pozisyon alması
4080 Çizgi bitiş koordinatı hesaplanması
Çizgi başlangıcı ve çizgi bitişi arasındaki hatın verilmesi
4090 Sayılar ve çizgilerin pozisyonu için döngü sonu
4100 Sonsuz döngü

10000 Trenin hızlandırılması ve yavaşlatılması için altprogram
10020 Ses şiddeti eğrisinin tarif edilmesi
10030 Gürültünün belirtilen kanal üzerine verilmesi
10040 Döngü gürültünün sona ermesini bekliyor
10050 Ana programa geri dönüş

20000 Trenin sabit hızı için alt program
20020 Ses şiddeti eğrisinin tarif edilmesi
20030 Gürültünün belirtilen kanal üzerine verilmesi
20040 Döngü gürültünün sona ermesini beklemektedir
20050 Ana programa geri dönüş

Sözlük:(Aşağıdaki programda kullanılan bazı kelimelerin anlamları) Monat=ay .Jahr=yıl, Tage= günler, Monatslange= ay uzunluğu,Wochentag= haftanın günü

3.3 Aylık Takvim

```
100 REM *** AYLIK TAKVİM ***
110 MODE 2:CLEAR:RESTORE
120 REM
130 REM *** Program listesi ***
140 PRINT STRING$(80,"_")
150 PRINT TAB(20);"* * *  AYLIK  TAK
VİM ***
160 PRINT STRING$(80,"_")
170 PRINT:PRINT:PRINT
180 PRINT" Bu program arzu edilen bir yılın
arzu edilen ayının haftalarını hesap
lar"
190 PRINT:PRINT
200 PRINT CHR$(7);:PRINT"lütfen ay numaras
ını ve yıl sayısını virgülle ayrılmış bir şe
kilde kaydediniz."
210 INPUT monat,jahr
220 IF monat<1 OR monat>12 THEN PRINT:PR
INT"Gregorien  Takvimi  sadece  bir  den  on
ikiye kadarki ayı tanır":GOTO 190
230 IF jahr<100 AND jahr>0 THEN  ahr=jah
r+1900:GOTO 250
240 IF jahr<1582 OR jahr<1583 AND monat<
11 THEN PRINT:PRINT"ilk olàrak 1582 senesi
Ekim ayında  gregoryen  takvimi  kullanılmış
tır. 4.10.1952  yılındaki  Juliyen  zaman  hesab
ını 15.10.1952  tarihli  gregoryer  zaman  hesa
bı izlemiştir":GOTO
190
250 REM
260 REM *** AY  isminin  ve  ay  uzunluğunun
okunması ***
```

```
270 FOR lauf=1 TO 12
280 READ a$,a
290 IF lauf=monat THEN monat$=a$:monatsl
aenge=a
300 NEXT lauf
310 REM
320 REM *** artık yıl tashihi ***
330 IF jahr MOD 400=0 OR jahr MOD 4=0 AN
D jahr MOD 100<>0 THEN artıkyıl=1
340 IF schaltjahr=1 AND monat=2 THEN mon
atslaenge=29
350 REM
360 REM *** Haftanın Günlerinin  hesaplanma
sı ***
370 tage=jahr*365+INT(jahr/4)-INT(jahr/1
00)+INT(jahr/400)-365
380 FOR lauf=1 TO 12
390 READ monatstage
400 IF lauf=monat THEN tage=tage+monatst
age
410 NEXT lauf
420 IF monat<3 AND artıkyıl=1 THEN tag
e=tage-1
430 wochentag=tage-INT(tage/7)*7
440 IF wochentag=0 THEN wochentag=7
450 REM
460 REM *** takvim ifadesi ***
470 CLS:LOCATE 30,1:PRINT monat$;jahr
480 FOR spalte=1 TO 73 STEP 12
490 LOCATE spalte,3
500 READ wochentag$:PRINT TAB(spalte);wo
chentag$
510 NEXT spalte
520 FOR reihe=6 TO 21 STEP 3
530 FOR spalte=2 TO 74 STEP 12
540 LOCATE spalte,reihe
550 wochentag=wochentag-1
```

```
560 IF wochentag<1 THEN monatslaenge=mon
atslaenge-1:IF monatslaenge>=0 THEN datu
m=datum+1:PRINT datum
570 NEXT spalte
580 NEXT reihe
590 PRINT:PRINT:PRINT
600 PRINT CHR$(7);:PRINT "bir başka takvim
ifadesi (e/h)?"
610 a$=INKEY$:IF a$="j"THEN 110 ELSE IF
a$<>"n" THEN 610
620 END
1000 REM
1010 REM *** takvim verileri ***
1020 DATA OCAK,31,ŞUBAT,28,MART,31,NİSAN,30,
MAYIS,31,HAZİRAN,30,TEMMUZ,31,AĞUSTOS,31
EYLÜL,30,EKİM,31,KASIM,30,
ARALIK,31
1030 DATA 1,32,60,91,121,152,182,213,244
,274,305,335
1040 DATA Pazartesi,Salı,Çarşamba,Perşembe,Cu
ma,Cumartesi,Pazar
```

100 program ismi
110 ön yükleme
140'dan
160'a kadar program isminin kaydedilmesi

170'den
200'e kadar metin ifadesi

210 Ay ve yılın kaydedilmesi
220 Ay'ın kontrol edilmesi
230 Yılın yüz yılının eksik olup olmadığının kontrol edilmesi
240 Yılın kontrol edilmesi

270 Ay ismi ve ay uzunluğunun okunması için döngü başlangıcı
280 Ay ismi ve ay uzunluğunun okunması
290 Aranan ismin ve aranan uzunluğun belirtilmesi
300 Ay ismi ve ay uzunluğunun okunması için döngü sonu

330 Artık yıl kontrolü
340 Artık yıl ve arzu edilen ay Şubat'ta
Ay uzunluğunun 29 güne uzatılması
370 Arzu edilen yıla kadar geçen günlerin hesaplanması

380 Arzu edilen aya kadar geçen günlerin okunması için döngü başlangıcı
390 Günlerin sayısının kaydedilmesi
400 Aranan günlerin belirtilmesi
410 Arzu edilen aya kadar geçen günlerin okunması için döngü sonu
420 Artık yıl ve arzu edilen ay Ocak ve Şubat'ta gün sayısının düzeltilmesi

430 Günlerin 7'ye bölünmesinde arta kalanın hesaplanması
440 Pazar için arta kalan günün düzeltilmesi
470 Ekranın silinmesi, yıl ve ayın verilmesi

480 Haftanın günlerini basmak için döngü başlangıcı
490 Kursörün pozisyon alması
500 Haftanın günlerinin okunması
510 Haftanın günlerinin basılması için döngü sonu

520 Kursörün yatay pozisyon alması için döngü başlangıcı
530 Kursörün dikey pozisyon alması için döngü başlangıcı
540 Kursörün pozisyon alması
550 Verilmesi gereken boş pozisyonun bire indirgenmesi
560 Bütün boş pozisyonlarının verilmesi anında, ay uzunluğunun bire indirgenmesi
Ay uzunluğunun sıfırdan büyük olması halinde, tarihin bir yükseltilmesi ve tarihin verilmesi
570 Kursörün dikey pozisyon alması için döngü sonu
580 Kursörün yatay pozisyon alması için döngü sonu

590'dan
600'e kadar metin ifadesi

610
620 Program sonu

3.4 Bioritm

Sözlük: Gebjahr=Doğum yılı,Monat=Ay,Jahr=yıl, Tage=günler, Monatslange=ay uzunluğu, Gebtage= gün uzunluğu, Monatschaft=ayın günleri, Gebmonat =doğum ayı, Spalte=sütun,Leange= uzunluk anlamlarına gelmektedir

```
100 REM *** B İ O R İ T M ***
110 MODE 2:CLEAR:RESTORE:DEG
120 REM
130 REM *** Programmenü ***
140 PRINT STRING$(80,"_")
150 PRINT TAB(22);"*** B İ O R İ T M
***
160 PRINT STRING$(80,"_")
170 PRINT:PRINT:PRINT
180 PRINT" Bu program arzu edilen bir tarih
için (doğum tarihinize göre) sizin için
şahsi bioritminizi hesaplar."
190 PRINT:PRINT
200 REM
210 REM *** ilk soruşturma ***
220 PRINT CHR$(7);:INPUT"doğum tarihinizi
virgülle ayrılmış bir şekilde gün,ay ve
yıl olarak kaydediniz"; doğum günü,doğum
ayı, doğum yılı
230 REM
240 REM *** bilgi işleme ***
250 ay=doğum ayı:yıl=doğum yılı
260 GOSUB 1000
270 doğum artık yılı=artık yıl:doğum ayı$=
ay$
280 doğum günü <1 OR doğum günü >ay uzunluğu
OR doğum ayı < 1 OR doğum ayı><12 OR doğum
```

```
yılı<>1 OR doğum yılı  2000 THEN PRINT"Doğum
tarihinizi biliyormusunuz?"= GOTO 190
290 IF gebjahr<100 THEN gebjahr=gebjahr+
1900
300 PRINT:PRINT
310 REM
320 REM  *** ikinci soruşturma ***
330 PRINT CHR$(7);:INPUT"beyanını istediği
niz tarihin ay numarasını ve yıl sayısını vi
roül  ile ayrılmış olarak kaydediniz.

340 REM
350 REM  *** bilgi işleme ***
360 IF  ay   <1 OR  ay   >12 THEN PRINT:PR
INT "Gregorien  takvimi  sadece  bir  den ıb
12'ye kadarki ayı tanır":GOTO 300
370 IF jahr<100 THEN jahr=jahr+1900
380 IF jahr<gebjahr OR jahr=gebjahr AND
monat<doğumayı  THEN PRINT:PRINT "Doğumdan
önceki  hayatın ihtimal harici olduğu size
açıklanma lıdır":GOTO 300

390 GOSUB 1000
400 REM
410 REM *** farkın gün olarak hesaplanması
***
420 tage=jahr*365+INT(jahr/4)-INT(jahr/1
00)+INT(jahr/400)
430 gebtage=gebjahr*365+INT(gebjahr/4)-I
NT(gebjahr/100)+INT(gebjahr/400)
440 FOR lauf=1 TO 12
450 READ monatstage
460 IF lauf=monat THEN tage=tage+monatst
age
470 IF lauf=gebmonat THEN gebtage=gebtag
e+monatstage
480 NEXT lauf
```

```
490 IF monat<3 AND  : artıkyıl =1 THEN tag
e=tage-1
500 IF gebmonat<3 AND doğumartıkyılı=1  TH
EN gebtage=gebtage-1
510 gebtage=gebtage+gebtag
520   farkr . .=tage-gebtage
530 REM
540 REM *** Ekran Metni ***
550 CLS:PRINT monat$:PRINT jahr
560 PRINT:PRINT:PRINT"1.eğri    . ":PRINT"
fiziki    "
570 PRINT:PRINT:PRINT" 2.eğri    ":PRINT"e
psikolojik"
580 PRINT:PRINT:PRINT" 3. eğri  .":PRINT"
manevi"
590 PRINT:PRINT:PRINT"Doğumgünü":PRINT
600 IF gebtag=1 OR gebtag=6 OR gebtag>19
 THEN PRINT gebtag;"ster" ELSE PRINT geb
tag;"ter"
610 PRINT gebmonat$:PRINT gebjahr
620 REM
630 REM *** Koordinat çizgileri ***
640 ORIGIN 116,80
650 DRAW 0,320:MOVE -8,150:DRAW 520,150
660 MOVE -8,0:DRAW 520,0:MOVE -8,300:DRA
W 520,300
670 LOCATE 12,2:PRINT"+1":LOCATE 12,11:P
RINT" 0":LOCATE 12,20:PRINT"-1"
680 FOR x=0 TO 496 STEP 16
690 MOVE x,0:DRAW x,-8
700 NEXT x
710 zaehler=1
720 FOR spalte=16 TO 76 STEP 4
730 LOCATE spalte,22:PRINT zaehler:zaehl
er=zaehler+2
740 NEXT spalte
750 REM
```

```
760 REM *** Eğrilerin kaydedilmesi ***
770 ORIGIN 116,230
780 alan=32
790 FOR laenge=23 TO 33 STEP 5
800 MOVE 0,0
810 index=(differenz-INT(differenz/laeng
e)*laenge)/laenge*360
820 IF jahr=gebjahr AND monat=gebmonat T
HEN    alan=32-gebtag:index=0:start=16*g
ebtag:MOVE start,0
830 FOR lauf=0 TO  ALAN  /laenge*360
840 DRAW lauf*laenge/22.5+start,150*SIN(
lauf+index)
850 NEXT lauf
860 NEXT laenge
870 LOCATE 1,25:PRINT CHR$(7);:PRINT" Ba
şka bir bioritm ifadesi istermisiniz(e/h)?"

880 a$=INKEY$:IF a$="j" THEN 100 ELSE IF
 a$<>"h" THEN 880
890 END
1000 REM
1010 REM *** Ay uzunluğu ve artık yıl testi
alt programı ***
1020 RESTORE
1030 FOR lauf=1 TO 12
1040 READ a$,a
1050 IF lauf=monat THEN monat$=a$:monats
laenge=a
1060 NEXT lauf
1070 IF jahr MOD 400=0 OR jahr MOD 4=0 A
ND jahr MOD 100<>0 THEN artıkyıl=1
1080 IF  artıkyıl=1  AND monat=2 THEN mo
natslaenge=29
1090 RETURN
10000 REM
10010 REM *** takvim bilgileri ***
```

```
1020 DATA OCAK,31,ŞUBAT,28,MART,31,NİSAN,30,
MAYIS,31,HAZİRAN,30,TEMMUZ,31,AĞUSTOS,31,
EYLÜL,30,EKİM,31,KASIM,30,ARALIK,31

10030 DATA 1,32,60,91,121,152,182,213,24
4,274,305,335
```

100 Program ismi
110 Ön yükleme

140'dan
160'a kadar program isminin kaydedilmesi

170'den
190'a kadar metin ifadesi

220 Doğum tarihinin kaydedilmesi
250 Alt program için değişkenlerin gösterilmesi
260 Alt programın çağrılması
270 Alt programdan verilerin belirtilmesi
280 Doğum tarihinin araştırılması
290 Doğum yılının yüzyıl ifadesinin eksik olup olmadığının araştırılması
330 Arzu edilen ay ve yılın kaydedilmesi
360 Ayın araştırılması
370 Yılın yüzyıl ifadesinin eksik olup olmadığının araştırılması
380 Arzu edilen ayın doğum ayı önünde olup olmadığının araştırılması
390 Alt programın çağrılması

420 Arzu edilen yıla kadar geçen günlerin hesaplanması
430 Doğum yılına kadar geçen günlerin hesaplanması

440 Arzu edilen aya ve doğum ayına kadar geçen günlerin okunması için döngü başlangıcı
450 Gün sayılarının okunması
460 Arzu edilen ay için aranılmış olan günlerin belirtilmesi

470 Doğum ayı için aranılmış olan günlerin belirtilmesi
480 Arzu edilen ay ve doğum ayına kadar geçen günlerin okunması için döngü sonu
490 Artık yıl durumunda ve Ocak yada Şubat ayında gün sayılarının düzeltilmesi
500 Artık yıl durumunda ve doğum ayı Ocak yada Şubat ayında gün sayılarının düzeltilmesi
510 Doğum gününe kadar geçen günlerin belirtilmesi
520 Doğum günü ile arzu edilen ay ve yıl arasındaki farkın hesaplanması
550 Ekranın silinmesi, ay ve yılın kaydedilmesi
560'dan
590'a kadar metin ifadesi

600 Doğum gününün kaydedilmesi
610 Doğum ayı ve yılının kaydedilmesi

640 Koordinat sıfır noktasının yerleştirilmesi
650 y ve x eksenlerinin kaydedilmesi
660 Sınır çizgilerinin kaydedilmesi
670 y eksen yazısının basılması

680 x eksen işaretlerinin yerleştirilmesi için döngü başlangıcı
690 İşaretlerin yerleştirilmesi
700 x eksen işaretlerinin yerleştirilmesi için döngü sonu

710 x ekseni değerinin belirtilmesi

720 x ekseni yazılarının basılması için döngü başlangıcı
730 Yazıların basılması
Yazı değerinin bire yükseltilmesi
740 x ekseni yazılarının basılması için döngü sonu

770 Koordinat sıfır noktasının yerleştirilmesi
780 Baskı alanının belirtilmesi
790 Üç eğrinin basılması için döngü başlangıcı
800 Grafik kursörünün pozisyon alması
810 Ay başlangıcı için sinüs eğrisi durumunun hesaplanması
820 Baskı alanının, sinüs eğrisi durumunun düzeltilmesi
Doğum ayı için eğri baskısı durumunda grafik kursörünün yeni pozisyon alması ve sinüs salınımının başlangıç noktasının belirtilmesi
830 Bir eğrinin basılması için döngü başlangıcı
840 Salınım uzunluğu dahil olarak x-koordinatının hesap-

lanması. Ay başlangıcı için sinüs eğri durumunu göz önüne alınarak y-koordinatının hesaplanması
850 Bir eğrinin baskısı için döngü sonu

860 Üç eğrinin baskısı için döngü sonu

870 Metnin basılması
880 Evet-hayır soruşturması
890 Program sonu

1000 Ay Uzunluğu ve artık yıl testi alt programı
1020 DATA işaretinin yerleştirilmesi
1030 Ay ismi ve ay uzunluğunun okunması için döngü başlangıcı
1040 Ay ismi ve ay uzunluğunun okunması
1050 Arzu edilen isim ve arzu edilen ayın belirtilmesi
1060 Ay ismi ve ay uzunluğu için döngü sonu

1070 Artık yıl araştırılması
1080 Artık yıl ve Şubat ayı durumunda ay uzunluğunun 29 güne uzatılması
1090 Ana programa geri dönüş

3.5 Finans Programı

```
100 REM *** FİNANS HESAPLAMASI ***
110 MODE 2:CLEAR
120 REM
130 REM *** Program Listesi ***
140 PRINT STRING$(80("80,"_")
150 PRINT TAB(15);"*** F İ N A N S  H E S A P L A M A S I ***"
160 PRINT STRIN$(80,"_")
170 PRINT:PRINT:PRINT
180 PLRINT"Hesaba bir defalık tahsilat";TAB(60);"(1)":PRINT:PRINT
190 PRINT"Yıl başlangıcında düzenli yıllık tahsilat";TAB(60);"(2)":PRINT:PRINT
200 PRINT" Ay başlangıcında düzenli aylık tahsilat";TAB(60);"(3)":PRINT:PRINT
210 PRINT"Bir kredinin amortisman süresi";
```

```
TAB(60);"(4)":PRINT:PRINT
220  PRINT"Lütfen  arzu  edilen  fonksiyonunun
numarasını kaydediniz "
230 GOSUP 3000
240funktion=a:ON funktion GOSUB     1000,1000,
1000,2000
250  CLS:LOCATE  15,13:PRINT  CHR$(7);:PRINT"
    Başka hesaplamalar istermisiniz? (e/h)"
260 a$=INKEY$"e" THEN 100 ELSE IF a$<>"h"THEN
260
270 END
1000 REM
1010  REM  ***  Fonksiyonların  işlemesi  1-3
***
1020 CLS:PRINT"Aşağıdaki hesaplar daha fazla
değişkenler gerektirir "
1030 PRINT"Değişkenlerden biri hesaplanabilir,
diğer parametreler tarafınızdan kaydedilmeli
dir. "
1040 PRINT"Lütfen hangi parametrenin hesaplan
ması gerektiğini belirtiniz "
1050 PRINT:PRINT:PRINT
1060 IF funktion=1 THEN PRINT"başlangıç";:
GOTO 1090
1070 IF funktion=2 THEN PRINT" yıllık tahsilat
";:GOTO 1090
1080 PRINT "aylık tahsilat";
1090 PRINT TAB(30);"(1)":PRINT:PRINT
1100 PRINT"Vade";TAB(30);"(3)":PRINT:PRINT
1100  PRINT"yüzde  faizler";TAB(30);"(2)":PRINT
:PRINT
1110 PRINT "Vade";TAB(30);"(3)":PRINT:PRINT
1120 PRINT " son kapital"; TAB(30);"(4)":
PRINT:PRINT
1130 GOSUP 3000
1140  IF  (funktion=2  OR  funktion=3)  AND  a=2
THEN LOCATE 40,11:PRINT"hesap mualesef mümkün
```

```
değil":GOTO 1220
1150 IF a <>1 THEN LOCATE 40,8:PRINT CHR$(7);:
INPUT kapital ELSE position=8
1160 IF a<>2 KPAH YDÇÜKA 40,11:PRINT CHR$(7)
;:INPUT faizler ELSE POSİTİON=11
1170 IF a<>3 THEN LOCATE 40,14:PRINT CHR$(7);:
INPUT vade ELSE position=14
1180 IF a<>4 THEN LOCATE 40,17:PRINT CHR$(7);:
INPUT son kapital ELSE position=17
1190 funktion=1 THEN GOSUP 4000:GOTO 1220
1200 IF funktion=2 THEN GOSUB:GOTO
1220
1210 GOSUB 6000
1220 LOCATE 1,25:PRINT CHR$(7);:PRINT
"Bu fonksiyonun bir tekrarını arzu
edermisiniz (e/h)?"
1230 a$=INKEY$:IF a$="e" THEN 1000
ELSE IF a<> "h" THEN 1230
1240 RETURN
2000 REM
2010 REM *** Fonksiyonun 4 işleyişi**
*
2020 CLS:ZONE 25:PRINT"kredi yüksekliğ
i","kredi faizi","aylık hisse(taksit)"
2030 LOCATE 15,1:PRINT CHR$(7);:INPUT
kredi:LOCATE 40,1:PRINT CHR$(7);:INPUT
faiz:LOCATE 65,1:PRINT CHR$(7);:INPUT
hisse
200 kredi=INT(kredi*100+0.5)/100:faiz=
INT(faiz*100+0.5)/100:hisse=INT(hisse
*100+0.5)/100
2050 PRINT:PRINT:ZONE 15:monat(ay)=0
2060 PRINT"Ay","Ay hissesi","Kredi fa
izleri","Amortisman"," Bakiye":PRINT
2070 Ay=Ay+1:faizler=kredi*faiz/1200:a
mortisman=hisse-faizler:kredi=kredi-
```

```
amortisman
2080 faizler=INT(faizler*100+0.5)/100:
amortisman=INT (amortisman*100+0.5)/10
0:kredi=INT(kredi*100+0.5)100
2090 PRINT ay,hisse,faizler,amortisman
,kredi
2100 IF>kredi 0 THEN 2070 ELSE PRINT
2110 Yıllar=INT(ay/12):aylar=ay-yıllar
*12
2120 PRINT "gerekir";yıllar: IF yıllar
>1 OR yıllar=0 THEN PRINT "yıllar";
ELSE PRINT"yıl";
2130 PRINT"ve"; aylar;IF aylar>1 OR
aylar=0 THEN PRINT "aylar"; ELSE PRINT
"ay";
1140 PRINT" Kredinizin taksitle ödenme
si için ":PRINT
2150 PRINT CHR$(7);:PRINT"Bu fonksiyon
un tekrarını istermisiniz (e/h)?"
2160 a$=INKEY$:IF a$="e" THEN 2000 ELSE
IF a$<>"h" THEN 2160
2170 RETURN
3000 REM
3010 REM *** Rakam kaydetme alt progra
mı ***
3020 PRINT CHR$(7);
3030 a$=INKEY$:a=VAL(a$):IF a 0>AND a<5
THEN RETURN ELSE 3030
4000 REM
4010 REM ***Fonksiyon 1 için formüller
***
4020 IF a=1 THEN sermaye=son sermaye/(
(1+faizler/100)^vade):baskı=sermaye:GOT
O 7000
4030 IF a=2 THEN faizler=100*((son ser
maye/sermaye)^(1/vade)-1):baskı=faizler
:GOTO 7030
```

```
400 IF a=3 THEN vade=LOG(son sermaye/
sermaye)/LOG (1+faizler/100):baskı=va
de:GOTO 7040
4050 son sermaye=sermaye*(1+faizler/10
0)^vade:baskı=son sermaye:GOTO 7050
5000 REM
5010 REM *** Fonksiyon 2 için formülle
r***
5020 IF a=1 THEN sermaye=son sermaye/(
((1+faizler/100)^(vade+1)-1)/(faizler/1
00)-1):GOTO 7000
5030 IF a=3 THEN VADE=LOG ((faizler/10
0)*(son sermaye/sermaye)=(1+faizler/10
0))/LOG (11+faizlerl/100)-1:GOTO 7040
5040 son sermaye=sermaye*(((1+faizler/
100)^(vade+1)-1)/(faizler/100)-1):GOTO
7050
6000 REM
6010 REM *** Fonksiyon 3 için formülle
r ***
6020 IF a=1 THEN sermaye=son sermaye/(
(12+78*faizler/1200)*(((1+faizler/100)
^vade-1)/(faizler/100))):GOTO 7000
6030 IF a=3 THEN vade=LOG ((faizler/10
0)*(son sermaye/(sermaye*(12+78*faizle
r/1200)))+1)/LOG(1+faizler/100):GOTO7040
6040  son  sermaye=sermaye*(12+78*faiz
ler/1200)*(((faizler/100)^vade-1)/(fai
zler/100)):GOTO 7050
7000 REM
7010 REM *** SONUÇ BASKISI ***
7020 sonuç=sermaye:GOTO 7060
7030 sonuç=faizler:GOTO 7060
7040 sonuç=vade:GOTO 7060
7050 sonuç=son sermaye
7060   LOCATE=40,pozition:sonuç=INT(so
nuç*100+0.5)/100:PRINT sonuç
7070 RETURN
```

100 Program ismi
110 Ön yükleme

140'dan
160'a kadar program isminin kaydedilmesi

170'den
220'ye kadar liste metninin basılması

230 "Rakam kaydetme" alt programının çağrılması
240 Arzu edilen alt programa geçiş

250 Ekranın ve metin ifadesinin silinmesi
260 Soruşturma durumu
270 Program sonu

1000 Fonksiyon 1, fonksiyon 2 ve fonksiyon 3 için alt alt program
1020'den
1120'ye kadar liste metninin basılması

1130 "Rakam kaydetme" alt programının çağrılması
1140 Hesaplama için formül olmadığı zaman metin basılsılması
1150'den
1180'e kadar zaruri bilgilerin kaydedilmesi
1190'dan
1210'a kadar arzu edilen fonksiyonun alt programının çağrılması

1220 Metnin basılması
1230 Soruşturma durumu
1240 Ana listeye geri dönüş

2000 Fonksiyon 4 için alt program
2020 Metnin basılması
2030 Zaruri bilgilerin kaydedilmesi
2040 Değerlerin yuvarlatılması
2050'den
2060'a kadar tablo başının kaydedilmesi
2070'den
2080'e kadar ayın bire yükseltilmesi ve bu ayın değerleririnin hesaplanması
2090 Değerlerin kaydedilmesi
2100 Amortisman müddetinin hesaplanması

2110'dan
2140'a kadar amortisman müddetinin basılması

2150 Metnin basılması
2160 Soruşturma durumu
2170 Ana listeye geri dönüş
3000 Rakam kaydetme alt programı
3030 Rakamların soruşturulması ve rakamlar 0 ile 5 arasında olunca ana listeye geri dönüş

4000 Fonksiyon 1 için formüller alt programı
4020'den
4050'ye kadar arzu edilen formüllerin seçimi, aranılan değerin hesaplanması ve sonuç bölümüne atlama

5000 Fonksiyon 2 için formüller alt programı
5020'den
5040'a kadar arzu edilen formülün seçimi, aranılan değerin hesaplanması ve sonuç bölümüne atlama

6000 Fonksiyon 3 için formüller alt programı
6020'den
6040'a kadar arzu edilen formülün seçimi, aranılan değerin hesaplanması ve sonuç bölümüne atlama

7000 Sonuç kayıt bölümünün başlangıcı
7020'den
7050'ye kadar sonuç için aranılan değerin belirtilmesi
7060 Yuvarlatma işlemi ve sonucun kaydedilmesi
7070 Fonksiyon 1, 2 ve 3 için alt programa geri dönüş

3.6 Fonksiyon Grafiği

```
100 REM *** FONKSİYON GRAFİĞİ ***
110 MODE 2:CLEAR
120 DEF FN fonksiyon(x)=x^3-x^2+3*x
130 xmini=-10:xmaxi=10
140 REM
150 REM
160 PRINT STRING$(80,"_")
170 PRINT TAB(20);"* * *  F O N K
S İ Y O N G R A F İ Ğ İ  * * *"
```

```
180 PRINT STRING$(80,"_")
190 PRINT:PRINT
200 PRINT"Bu program verilen fonksiyon
için grafik çizer."
210 PRINT"(Y-Ekseni için asimptot
paraleli çizilir.)":PRINT
220 PRINT"Aşağıdaki ihtimaller seçim
içindir:":PRINT:PRINT
230 PRINT"Yeni fonksiyon kaydediniz";
TAB(30);"(1)":PRINT
240 PRINT"x-Ekseni sınırlarını tespit
ediniz";TAB(30);"(2)":PRINT
250 PRINT"y-Ekseni sınırlarını tespit
ediniz";TAB(30);"(3)":PRINT
260 PRINT"Fonksiyon grafiği kaydediniz
"TAB(30);"(4)":PRINT:PRINT:PRINT
270 PRINT CHR$(7);:PRINT"Lütfen arzu
ettiğiniz bir numarayı kaydediniz "
280 a$=INKEY$:a=VAL(a$):IF a<1 OR a>4
THEN 280
290 CLS:ON a GOSUB 1000,2000,3000,4000
300 CLS:IF a=4 THEN LOCATE 12,13:PRINT
CHR$(7);:PRINT"Bu program devam etme-
li midir? (e/h) ?" ELSE 150
310 a$=INKEY$:IF a$="e" THEN CLS:GOTO
150 ELSE IF a$<>"h" THEN 310
320 END
1000 REM
1010 REM *** Yeni fonksiyon kaydediniz
***
1020 PRINT"Lütfen fonksiyonunuzu
aşağıdaki şekilde 120'inci satıra
kaydediniz:":PRINT
1030 PRINT"120 DEF FN fonksiyon(x)=x^4
+2*x^3-x ..... vb.":PRINT
1040 PRINT"Buna müteakip programı
RUN komutu ile tekrar başlatabilirsi-
```

```
niz.":PRINT
1050 PRINT"Program değişikliği yapma-
mış iseniz, Program CONT komutuyla
devam edilebilir":PRINT:PRINT:STOP
1060 RETURN
2000 REM
2010 REM *** x-Ekseni sınırlarını
tespit ediniz ***
2020 PRINT"x-Ekseninin alt ve üst
sınırlarını kaydediniz ":PRINT:PRINT
2030 PRINT CHR$(7);:INPUT"alt sınır";x
mini:PRINT
2040 PRINT CHR$(7);:INPUT"üst sınır";x
maxi:PRINT
2050 IF xmaxi<=xmini THEN PRINT"hata-
lı kayıt ":PRINT:GOTO 2020
2060 RETURN
3000 REM
3010 REM *** y-Ekseni sınırlarını
tespit ediniz ***
3020 PRINT"y-Ekseninin alt ve üst
sınırlarını kaydediniz ":PRINT:PRINT
3030 PRINT"(Şayet iki kere sıfır
kaydederseniz, bilgisayar sınırları
kendisi hesaplar.)":PRINT:PRINT
3040 PRINT CHR$(7);:INPUT"alt sınır";y
mini:PRINT
3050 PRINT CHR$(7);:INPUT"üst sınır";y
maxi:PRINT
3060 IF y mini=0 AND ymaxi=0 THEN
3070 ELSE IF ymaxi<=ymini THEN PRINT"
hatalı kayıt ":PRINT:GOTO 3020
3070 RETURN
4000 REM
4010 REM *** Eksen sınırlarını hesap-
layınız ***
```

```
4020 xfark=xmaxi-xmini
4030 xadım=639/xfark
4040 IF ymini<>0 OR ymaxi<>0 THEN
4400 ELSE flag=1
4200 REM
4210 REM *** Kaydetmediyseniz, y-Eksen
sınırlarını hesaplayınız ***
4220 LOCATE 10,13:PRINT"y-Eksen sınır
ları hesaplanıyor "
4230 FOR x=xmini TO xmaxix STEP xadım^
-1
4240 y=FN fonksiyon(x)
4250 IF y mini>y THEN ymini=y
4260 IF ymaxi<y THEN ymaxi=y
4270 NEXT x
4400 REM
4410 REM *** Koordinatın sıfır nokta-
sını hesaplayınız ***
4420 yfark=ymaxi-ymini
4430 yadım=399/yfark
4440 xeksen=xmini/-xfark*639
4450 yeksen=ymini/-yfark*399
4460 CLS:ORIGIN xeksen, yeksen
4600 REM
4610 REM *** Koordinat hatlarını
çiziniz ***
4620 MOVE xmini*xadım,0
4630 DRAW xmaxi*xadım,0
4640 MOVE 0,ymini*yadım
4650 DRAW 0,y maxi*yadım
4660 ymini=INT(ymini*1000+0.5)/1000
4670 ymaxi=INT(ymaxi*1000+0.5)/1000
4680 ykursör=INT((-yeksen+399)/16+1)
4690 IF yeksen<0 THEN ykursör=25 ELSE
IF yeksen>399 THEN ykursör=1
4700 LOCATE 1,ykursör:PRINT INT(xmaxi*
1000+0.5)/1000;
```

```
4710 LOCATE 80-LEN(STR$(INT(xmaxi*1000
+0.5)/1000)),ykursör:PRINT INT(xmaxi*
1000+0.5)/1000;
4720 IF LEN(STR$(ymini)) LEN(STR$(y
maxi)  THEN  uzunluk=LEN(STR$(ymini))
ELSE uzunluk=LEN(STR$(ymaxi))
4730 xkursör=INT(xeksen/8+1)-uzunluk
4740 IF xkursör<1 THEN xkursör=1
ELSE IF xeksen>639 THEN xkursör=80-
uzunluk
4760 LOCATE xkursör,25:PRINT ymini;
4770 IF flag=1 THEN ymini=0:ymaxi=0:
flag=0
4800 REM
4810 REM *** fonksiyon grafiği veriniz
***
4820 ypozisyon=FN fonksiyon (xmini)*
4830 IF ypozisyon>32767 THEN ypozisyon
=32767  ELSE IF ypozisyon<-32768
THEN pozisyon=-32768
4840 MOVE xmini*xadım,ypozisyon
4850 FOR x=xmini TO xmaxi STEP xadım^
-1
4860 y=FN fonksiyon(x)
4870 ypozisyon=y*yadım
4880 IF ypozisyon<32768 AND ypozisyon
>-32769 THEN DRAW x*xadım,ypozisyon
4890 NEXT x
4900 PRINT CHR$(7);
4910 a$=INKEY$:IF a$="" THEN 4910
4920 RETURN
```

100 Program ismi
110 Ön yükleme
120 Fonksiyonun tarif edilmesi
130 x ekseni için standart sınırların belirtilmesi

160'dan
180'e kadar program isminin kaydedilmesi
190'dan
270'e kadar liste metninin basılması

280 Arzu edilen ihtimalin soruşturulması
290 Ekranın silinmesi ve arzu edilen alt programa geçiş
300 Ekranın ve metinin silinmesi
310 Soruşturma durumu
320 Program sonu

1000 Yeni bir fonksiyonun kaydı için alt program
1020'den
1050'ye kadar metnin baskısı ve programın sona ermesi
1060 Listeye geri dönüş

2000 x ekseni sınırlarının tespit edilmesi için alt program
2020 Metnin basılması
2030 Alt sınırların kaydedilmesi
2040 Üst sınırların kaydedilmesi
2050 Hatalı kayıt denemesi
2060 Listeye geri dönüş

3000 y ekseni sınırlarını tespit etmek için alt program
3020'den
3030'a kadar metnin basılması
3040 Alt sınırların kaydedilmesi
3050 Üst sınırların kaydedilmesi
3060 Hatalı kayıt denemesi
3070 Listeye geri dönüş

4000 Fonksiyon grafiği verilmesi için alt program
4020 x ekseni uzunluğunun hesaplanması
4030 x birimi için adım büyüklüğünün hesaplanması
4040 y eksen sınırlarının belirtilip belirtilmediği yada hesaplalanmasının gerekli olup olmadığının (flag=1) kontrolü

4220 Metnin basılması

4230 x eksen sınırları içersinde y'nin en küçük ve en büyük değerinin hesaplanması için döngü başlangıcı
4240 y'nin hesaplanması
4250 y için en küçük değerin olup olmadığının denenmesi
4260 y için en büyük değerin olup olmadığının denenmesi

4270 x ekseni sınırları içersinde y'nin en küçük ve en büyük değerinin hesaplanması için döngü sonu

4420 y ekseni uzunluğunun hesaplanması
4430 y birimi için adım büyüklüğünün hesaplanması
4440 Koordinat sıfır noktasının x-koordinatının hesaplanması
4450 Koordinat sıfır noktasının y-koordinatının hesaplanması

4460 Ekranın silinmesi ve koordinat sıfır noktasının yerleştirilmesi

4620 Grafik kursörünün pozisyon alması
4630 x ekseninin verilmesi
4640 Grafik kursörünün pozisyon alması
4650 y ekseninin verilmesi

4660 Alt y eksen sınırının yuvarlatılması
4670 Üst y eksen sınırının yuvarlatılması

4680 Kursörün dikey pozisyonunun hesaplanması
4690 x ekseni ekran dışına taşarsa, dikey kursör pozisyonunun düzeltilmesi
4700 Yuvarlatılan alt x ekseni sınırının ekranın sol kenarında belirtilmesi
4710 Yuvarlatılan üst x ekseni sınırının ekranın sağ kenarında belirtilmesi

4720 y eksen sınır metninin uzunluğunun hesaplanması
4730 Kursörün yatay pozisyonunun hesaplanması
4740 y eksen sınırları ekran dışına taşıyorsa kursörün yatay pozisyonunun düzeltilmesi
4750 Üst y eksen sınırlarının ekranın üst kenarında belirtilmesi
4760 Alt y eksen sınırlarının ekranın alt kenarında belirtilmesi
4770 y eksen sınırları hesaplanınca (flag=1) geri yerleştirilmesi

4820 Alt x eksen sınırları için y koordinatının hesaplanması
4830 y koordinatının değeri integer alanında ise, hata ikazından kaçınmak için y koordinatının düzeltilmesi

4840 Grafik kursörünün pozisyon alması

4850 Fonksiyon grafiğinin belirtilmesi için döngü başlangıcı
4860 y'nin hesaplanması
4870 y koordinatının hesaplanması
4880 y koordinatının değeri integer alanı içinde ise grafik bölüm parçasının belirtilmesi
4890 Fonksiyon grafiğinin belirtilmesi için döngü sonu
4900 "READY" ikazı olarak sinyal sesi
4910 Bir tuş basılıncaya kadar, tuşların soruşturulması
4920 Listeye geri dönüş

3.7 Soruşturma Programı

```
100 REM *** SORUŞTURMA PROGRAMI ***
110 MODE 2:CLEAR:RANDOMIZE TIME
120 DIM veri$(1,1000),flag(1000)
130 REM
140 REM *** Program listesi ***
150 PRINT STRING$ (80,"_")
160 PRINT TAB(15);"* * *  S O R U Ş
T U R M A  P R O G R A M I  * * *"
170 PRINT STRING$(80,"_")
180 LOCATE 10,15:PRINT CHR$(7);:PRINT
"Bu program için bilgi ister misiniz
(e/h) ?"
190 a$=INKEY$:IF a$="e"THEN GOSUB
10000 ELSE IF a$<>"h" THEN 190
200 MODE 1:PRINT"İster misiniz:":PRINT
:PRINT
210 PRINT"veri kaydediniz,";TAB(30);"(
1)":PRINT
220 PRINT"verileri tanzim ediniz,":TAB
(30);"(2)":PRINT
230 PRINT"verileri siliniz",;TAB(30);
"(3)":PRINT
240 PRINT"verileri okuyunuz,";TAB(30);
"(4)":PRINT
250 PRINT"verileri okumayı tamamlayı-
```

```
nız,";TAB(30);"(5)":PRINT
260 PRINT"soruşturuldu";TAB(30);"(6)":
PRINT
270 PRINT:PRINT"bitsin mi?";TAB(30);"
(7):PRINT CHR$(7);
280 a$=INKEY$:a=VAL(a$):IF a<1 OR a>7
THEN 280
290 MODE 2:IF gösterge=0 AND a<>1
AND a<>4 AND a<>7 THEN LOCATE 20,13:
PRINT"Hiçbir veri yoktur ":FOR lauf=1
TO 2000:NEXT:GOTO 200
300 ON a GOSUB 1000,2000,3000,4000,
5000,6000,310:GOTO 200
310 LOCATE 25,13:PRINT"tekrar görüşmek
üzere "
320 END
1000 REM
1010 REM *** verilerin kaydedilmesi
***
1020 PRINT"Birbiriyle ilgili veri
alanlarını daima arka arkaya kaydedin
ki ekranda yanyana görülsünler."
1030 PRINT"Bir veri çiftini kaydeder-
ken yan ışıkla ENTER tuşuna basarsanız
kayıt işlemi tekrarlanır."
1040 PRINT"Şayet ENTER tuşuna iki kez
basarsanız, veri kayıt işlemi sona
erer.":PRINT
1050 gösterge=gösterge+1:PRINT göster-
ge
1060 LOCATE 20,VPOS(#0)-1:PRINT CHR$(
7);:INPUT a$:LOCATE 50,VPOS(#0)-1:
PRINT CHR$(7);:INPUT b$
1070 IF a$<>"" AND b$<>"" THEN veri$(0
gösterge)=a$:veri$(1,gösterge)=b$:GOTO
1050 ELSE IF a$<>"" OR b$<>"" THEN
1060
```

```
1080 gösterge=gösterge-1
2010 REM *** verilerin tanzim edilmesi
***
2020 PRINT CHR$(7);:INPUT"Lütfen
yanlış veri alanını kaydediniz";yanlış
$:PRINT:PRINT
2030 FOR veri cümlesi=0 TO 1
2040 FOR veri alanı=1 TO gösterge
2050 THEN PRINT CHR$(7);:INPUT"Düzelti
lenveri alanı nasıldır";doğru$:PRINT:
PRINT:veri$(veri cümlesi,veri alanı)=
doğru$:GOTO 2090
2060 NEXT veri alanı
2070 veri cümlesi
2080 PRINT"veri alanı";yanlış$;"mevcut
değil " PRINT:PRINT
2090 PRINT CHR$(7);:PRINT"diğer verile
ri tanzim etmek istiyor musunuz (e/h)
?":PRINT:PRINT
2100 a$=INKEY$:IF a$="e" THEN 2000
ELSE
IF a$<>"n" THEN 2100
2110 RETURN
3000 REM
3010 REM *** verilerin silinmesi ***
3020 PRINT CHR$(7);:PRINT"Lütfen silin
mesi gereken veri çiftini virgül
ile ayırarak kaydediniz "
3030 INPUT a$,b$:PRINT:PRINT
3040 FOR veri alanı=1 TO gösterge
3050 IF veri$(0,veri alanı)=a$ AND
veri$(1,veri alanı)=b$ OR veri$(0,veri
alanı)=b$ AND veri$(1,veri alanı)=a$
THEN 3080
3060 NEXT veri alanı
3070 PRINT"veri çifti";a$;", "b$;"mev
cut değil ":GOTO 3150
```

```
3080 FOR lauf=veri alanı TO=gösterge-1
3090 veri$(1,veri alanı)=veri$(0,veri
alanı+1)
3100 veri$(1,veri alanı)=veri(1,veri
alanı+1)
3110 NEXT lauf
3120 gösterge=göst erge-1
3130 PRINT"veri çifti";a$;", ";b$;"
silinmiştir "
3140 IF gösterge=0 THEN FOR warten=1
TO 2000:NEXT warten=GOTO 3170
3150 PRINT CHR$(7);:PRINT:PRINT:PRINT"
Diğer veri çiftlerini silmek ister mi
siniz (e/h) ?":PRINT:PRINT
3160 a$=INKEY$:IF a$="e" THEN 3000
ELSE IF a$  "h" THEN 3160
3170 RETURN
4000 REM
4010 REM *** Verilerin okunması ***
4020 PRINT CHR$(7);:INPUT"Okunması
gereken verinin ismi nasıldır"; isim$:
isim=UPPER$(isim$):PRINT
4030 PRINT"Kaseti okunması gereken
verinin başına kadar sarınız.
":PRINT
4040 OPENIN isim$
4050 INPUT#9,gösterge
4060 FOR veri cümlesi=0 TO 1
4070 FOR veri alanı=0 TO gösterge
4080 INPUT#9,veri$(veri cümlesi,veri
alanı)
4090 NEXT veri alanı
4100 NEXT veri cümlesi
4110 CLOSEIN
4120 RETURN
5000 REM
5010 REM *** Verilerin okunup bitiril
mesi ***
```

```
5020 isim$=STRING$(16," "):PRINT CHR$(
7);:INPUT"Okunması gereken verinin
ismi nasıldır";isim$:PRINT
5030 PRINT"Kaset üzerinde veri için
boş bir yer arayınız.":PRINT
5040 OPENOUT isim$
5050 PRINT#9,gösterge
5060 FOR veri cümlesi=0 TO 1
5070 FOR veri alanı=0 TO gösterge
5080 PRINT#9,veri$(veri cümlesi, veri
alanı)
5090 NEXT veri alanı
5100 NEXT veri cümlesi
5110 CLOSEOUT
5120 RETURN
6000 REM
6010 REM *** verilerin soruşturulması
***
6030 PRINT CHR$(7);:INPUT"Birinci mi
yoksa ikinci veri cümlesi mi soruştu
rulmalıdır (1/2) ";seçim:seçim=2-seçim
:=2-seçim:PRINT:PRINT
6030 PRINT CHR$(7);:INPUT"Genelde ne
kadar soruşturmalısınız";miktar:PRINT:
PRINT
6040 FOR soruşturma=1 TO miktar
6050 rastgele=INT(RND*gösterge+1):IF
flag(rastgele)=1 THEN 6050 ELSE flag
(rastgele)=1
6060 PRINT:PRINT soruşturma;TAB(20);ve
ri$(seçme,tesadüf);:LOCATE 60,VPOS(#0)
:PRINT CHR$(7);:INPUT cevap$
6070 IF cevap$="" THEN cevap$=error"
6080 IF cevap$=veri$(1-seçme,rastgele)
THEN PRINT"doğru ":GOTO 6110
6090 hata=hata+1:PRINT"Bu maalesef
yanlıştır "
```

```
6100 PRINT"Doğru cevap:";veri$(1-seçme
,rastgele)
6110 IF soruşturma/gösterge=INT(soruş
turma/gösterge) THEN GOSUB 20000
6120 NEXT soruşturma
6130 PRINT:PRINT:PRINT"Siz burada";
miktar;"soruşturma";hata;" çözümü
bilmiyorsunuz "
6140 GOSUB 20000:FOR warten=1 TO 2000:
NEXT warten
6150 RETURN
10000 REM
10010 REM *** Kullanıcı bilgi için bil
gi alt programı ***
1020 LOCATE 1,10:PRINT"Bu programı
öğrenme esnasında kuvvetlendirebilirsi
niz "
10030 PRINT"Birbirleri ile belirli bir
ilişkide bulunan iki veri cümlesinin
alanını belirtmek ve siz de diğer veri
cümlesininilgili alanını kaydetmek
suretiyle, bilgisayar sizi soruşturur.
":PRINT
10040 PRINT"Veri cümlelerinde değişik
dillerin kelimeleri, yabancı kelimeler
ve onların tarifleri, şekilleri,
ve onların işaretleri, yıl sayıları ve
tarihi olaylar vs. söz konusudur.":
PRINT
10050 PRINT"Verilerinizi her defasında
tekrar programın başına kaydetmemeniz
için, veri cümlelerini banda tamamen
okuyunuz ve daha sonra aynı şekilde
tekrar okuyabilirsiniz":PRINT
10060 PRINT"Hepsini okumuş iseniz, lüt
fen ENTER tuşuna basınız ":PRINT CHR$(
7);
```

```
10070 a$=INKEY$:IF a$  CHR$(13) THEN
10070 ELSE RETURN
20000 REM
20010 REM *** Bütün soruşturma pozis
yonlarının silinmesi ***
20020 FOR index=1 TO gösterge
20030 flag(index)=0
20040 NEXT index
20050 RETURN
```

100 Program ismi
110 Ön yükleme
120 Veri matrislerinin dimensiyonu
150'den
170'e kadar program isminin verilmesi
180 Metin baskısı
190 Kullanıcının soruşturulması, "Kullanıcı için bilgi" alt programının muhtemelen çağrılması
200'den
270'e kadar-liste metninin verilmesi

280 Arzu edilen fonksiyonun soruşturulması
290 Fonksiyonun çağrılması hatalı veriden dolayı geçersiz olunca, liste metninin verilmesi tekrarlanır
300 Arzu edilen alt programlar
310 Metin basımı
320 Program sonu

1000 Verilerin kaydedilmesi için alt sistem
1020'den
1040'a kadar metin basımı

1050 Veri göstergesinin bir yükseltilmesi ve göstergenin verilmesi
1060 Veri çiftinin kaydedilmesi
1070 Kayıt işleminin kontrolü, diğer muhtemel kayıtlar yada yeni kayıtlar
1080 Göstergenin bir indirgenmesi ile veri göstergesinin düzeltilmesi
1090 Listeye geri dönüş

2000 Verilerin tasnifi için alt sistem

2020 Yanlış veri alanının kaydedilmesi

2030 Veri cümlelerinin çevrimi için döngü başlangıcı
2040 Veri alanlarının çevrimi için döngü başlangıcı
2050 Yanlış veri alanının bulunması anında:
Doğru alanın kaydedilmesi ve yanlış alanın yerine geçmesi
Döngüden çıkış
2060 Veri alanının çevrimi için döngü sonu
2070 Veri cümlelerinin çevrimi için döngü sonu

2080 Arama döngüsünün başarısız sonuçlanması halinde metin basımı ve bir sonraki program satırına atlama

2090 Metin basımı
2100 Soruşturma programı
2110 Listeye geri dönüş

3000 Verilerin silinmesi için alt program
3020 Metin basımı
3030 Silinmesi gereken veri çiftinin kaydedilmesi
3040 Veri alanının çevrimi için döngü başlangıcı
3050 Silinmesi gereken veri çiftinin bulunması anında:
Döngüden çıkış
3060 Veri alanının çevrimi için döngü sonu
3070 Arama döngüsünün başarısız sonuçlanması halinde metin basımı ve daha sonraki döngüye atlama

3080 Veri çiftinin silinmesi gereken veri çiftinin yerine geçmesi için döngü başlangıcı
3090'dan
3100'e kadar veri çiftinin yerleşmesi
3110 Veri çiftinin silinmesi gereken veri çiftinin yerine geçmesi için döngü sonu

3120 Göstergenin bir indirgenmesi ile veri göstergesinin düzeltilmesi
3130 Metin basımı
3140 Son veri çifti silinince alt programın sona ermesi
3150 Metin basımı
3160 Soruşturma durumu
3170 Listeye geri dönüş

4000 Verilerin okunması için alt program

4020 Veri isminin kaydedilmesi
4030 Metin basımı
4040 Okuma verisinin açılması
4050 Veri göstergesinin okunması

4060 Veri cümlelerinin okunması için döngü başlangıcı
4070 Veri alanlarının okunması için döngü başlangıcı
4080 Bir veri alanının okunması
4090 Veri alanlarının okunması için döngü sonu
4100 Veri cümlelerinin okunması için döngü sonu

4110 Okuma verisinin kapatılması
4120 Listeye geri dönüs
5000 Verilerin okutulup bitirilmesi için alt program
5020 Veri isminin kaydedilmesi
5030 Metin basımı

5040 Okutma verisinin açılması
5050 Veri göstergesinin okutulması

5060 Veri cümlelerinin okunup bitirilmesi için döngü başlangıcı
5070 Veri alanlarının okunup bitirilmesi için döngü başlangıcı
5080 Bir veri alanının okutulması
5090 Veri alanının okunup bitirilmesi için döngü sonu
5100 Veri cümlelerinin okunup bitirilmesi için döngü sonu

5120 Okutma verisinin kapatılması

5120 Listeye geri dönüş

6000 Verilerin soruşturulması için alt program
6020 Soruşturulması gereken veri cümlesinin kaydedilmesi
6030 Soruşturma sayısının kaydedilmesi

6040 Verilerin soruşturulması için döngü başlangıcı
6050 Rastgele sayının üretilmesi ve sayının soruşturulup soruşturulmadığının kontrolü (flag=1)
6060 Veri alanının soruşturulması ve kaydedilmesi
6070 Boş kayıt durumunda düzeltme işlemi
6080'den
6100'e kadar veriler veri alanının kontrolü ve metin basımı

6110 Bütün veri alanları tam bir kez soruşturulduğunda, "Bütün soruşturma pozisyonlarının silinmesi" alt prog-

ramının çagrılması
6120 Verilerin soruşturulması için döngü sonu
6130 Hata sayısının verilmesi
6140 "Bütün soruşturma pozisyonlarının silinmesi" alt programının çağrılması ve bekleme döngüsü
6150 Listeye geri dönüş
10000 Kullanıcı için bilgi alt programı
10020'den
10060'a kadar metin basımı
10070 ENTER tuşuna basmak suretiyle listeye geri dönüş
20000 Bütün soruşturma pozisyonlarının silinmesi için alt program
20020 Bütün soruşturma pozisyonlarının çevrimi için döngü başlangıcı
20030 Bir pozisyonun silinmesi (flag=0)
20040 Bütün soruşturma pozisyonlarının çevrimi için döngü sonu
20050 Verilerin soruşturulması alt programına geri dönüş

3.8 Adres Yönetimi

```
100 REM *** ADRES YÖNETİMİ ***
110 CLEAR
120 maxi=100
130 DIM hitap$(maxi), ilkadı$(maxi),
adı$(maxi),cadde$(maxi),numara$(maxi)
posta kutusu$,posta kodu$(maxi),yer$(
maxi),kod numarası$(maxi),telefon$(
maxi)
140 REM
150 REM *** Program Listesi ***
160 MOD 2
170 PRINT STRING$(80"_")
180 PRINT TAB(15);"* * *  A D R E S
Y Ö N E T İ M İ  * * *"
190 PRINT STRING$(80"_")
200 PRINT:PRINT
```

```
210 PRINT"Adresi kaydediniz";TAB(30);"
(1)":PRINT
220 PRINT"Adresi tanzim ediniz";TAB(30
);"(2)":PRINT
230 PRINT"Adresi siliniz";TAB(30);"(3)
":PRINT
240 PRINT"Adresi okuyunuz";TAB(30);"(5
)":PRINT
250 PRINT"Adresi tamamlayınız";TAB(30)
(4)":PRINT
260 PRINT"Adresi basınız";TAB(30);"(6)
":PRINT
270 PRINT n,"Programı bitiriniz";TAB(3
0);"(70)":PRINT:PRINT
280  PRINT  CHR$(7);:PRINT"Arzu  edilen
fonksiyonun  numarasını  kaydediniz "
290  a$=INKEY$:a=VAL(a$):IF  a<1  OR
a>7 THEN 290
300  IF  gösterge=0  AND  a<> 4  AND  a<>7
THEN CLS:LOCATE 20,13:PRINT"Bir adres
vardır ":FOR bekleyiniz=1 TO 2000:NEXT
bekleyiniz:GOTO 140
310 ON a GOSUB 1000,2000,3000,4000,500
0,6000320:GOTO 140
320 CLS:LOCATE 25,13:PRINT"Görüşmek
üzere "
330 END
1000 REM
1010 REM *** Adres kaydı ***
1020 CLS:PRINT"Şimdi adresi kaydediniz
 "
1030 PRINT"bir noktayı beyan etmezseni
z  ya  da  beyan  etmek  istemezseniz,
kayıt yapmaksızın sadece ENTER tuşuna
basınız"
1040 gösterge=gösterge+1:index=göster
ge
```

```
1050 GOSUB 10000
1060 PRINT CHR$(7);:PRINT"diğer bir
adresi istiyor musunuz (e/h)?"
1070 a$=INKEY$:IF a$="e" THEN 10000ELSE
IF a$<>"h" THEN 1070
1080 RETURN
2000 REM
2010 REM *** Adres tanzimi ***
2020 CLS:PRINT CHR$(7);:INPUT"Adresi
değişen şahsın ismi nedir (ilk adı,
soyadı)";ilk adı$,adı$:PRINT:PRINT
2030 FOR index=1 TO gösterge
2040 IF ilk adı$=ilk adı$(index) AND
adı$=adı$(index) THEN 2070
2050 NEXT index
2060 PRINT"Adreste"ilk adı$;"";adı$"me
vcut değildir ":GOTO 2090
2070 PRINT"Adresde "ilk adı;" "adı$"
şöyledir:":GOSUB 10000
2080 PRINT CHR"Şimdi değişen adresi ka
ydediniz.":GOSUB 10000
2090  PRINT  CHR$(7);:PRINT:PRINT"Diğer
adresi değiştiriniz (e/h)?"
2100 a$=INKEY$:IF a$="e" THEN 2000
ELSE IF a$  "h" THEN 2100
2110 RETURN
3000 REM
3010 REM *** Adresin okunması ***
3020 CLS:PRINT CHR$(7);:INPUT"Adresi
silinen sahsın adı nedir (ilk adı,
soyadı)";ilk adı$,adı$:PRINT:PRINT
3030 FOR index=1 TO gösterge
3040 IF ilk adı$=ilk adı$(index) AND
adı$=adı$(index) THEN 3070
3050 NEXT index
3060 PRINT"Adreste "ilk adı$;" "adı$"
mevcut değildir  ":GOTO "3220
3070 FOR lauf=index TO gösterge-1
```

```
3080 hitap$(lauf)=hitap(lauf+1)
3090 ilk adı$(lauf)=ilk adı$(lauf+1)
3100 adı$(lauf)=adı$(lauf+1)
3110 cadde$(lauf)=cadde$(lauf+1)
3120 numara$(lauf)=numara$(lauf+1)
3130 posta kutusu$(lauf)=posta kutusu$
(lauf+1)
3140 posta kodu$(lauf)=posta kodu$(la
uf+1)
3150 yer$(lauf)=yer$(lauf+1)
3160 telefon kodu$(lauf)=telefon kodu$
(lauf+1)
3170 telefon$(lauf)=telefon$(lauf+1)
3180 NEXT lauf
3190 gösterge=gösterge-1
3200 PRINT"Adreste  "ilk adı$;" ";adı$
silindi  "
3210 IF gösterge=0 THEN FOR lauf=1 TO
2000:NEXT:GOTO 3240
3220 PRINT CHR$(7);:PRINT:PRINT"diğer
adresleri siliniz (e/h) ?"
3230 a$=INKEY:IF a$="e" THEN 3000 ELSE
IF a$<>"h" THEN 3230
3240 RETURN
4000 REM
4010 REM *** Adresin okunması ***
4020 CLS:PRINT CHR$(7);:PRINT"Kaseti
okunan adresin başlangıcına kadar sarı
nız."PRINT
4030 OPENİN "ADRES"
4040 INPUT#9,gösterge
4050 FOR index=1 TO gösterge
4060 INPUT#9,hitap$(index)
4070 INPUT#9,ilk adı$(index)
4080 INPUT#9,adı$(index)
4090 INPUT#9,cadde$(index)
4100 INPUT#9,numara$(index)
4110 INPUT#9,posta kutusu$(index)
```

```
4120 INPUT#9,posta kodu
4130 INPUT#9,yer$(index)
4140 INPUT#9,telefon kodu(index)
4150 INPUT#9;telefon$(index)
4160 NEXT index
4170 CLOSEIN
4180 RETURN
5000 REM
5010 REM *** Adresin tamamlanması ***
5020 CLS:PRINT CHR$(7);:PRINT"Kasette
adres için boş yer arayınız.":PRINT
5030 OPENOUT "ADRES"
5040 PRINT#9,gösterge
5050 FOR index=1 TO gösterge
5060 PRINT#9,hitap$(index)
5070 PRINT#9,ilk adı$(index)
5080 PRINT#9,adı$(index)
5090 PRINT#9,cadde$(index)
5100 PRINT#9,numara$(index)
5110 PRINT#9,posta kutusu$(index)
5120 PRINT#9,posta kodu$(index)
5130 PRINT#9,yer$(index)
5140 PRINT#9,telefon kodu$(index)
5150 PRINT#9,telefon$(index)
5160 NEXT index
5170 CLOSEOUT
5180 RETURN
6000 REM
6010 REM *** Adres beyanı ***
6020 MOD 2:PRINT"Hangi fonksiyonu iste
siniz ?":PRINT:PRINT:PRINT
6030 PRINT"Bütün adresleri ekrana bası
nız";TAB(60);"(1)":PRINT
6040 PRINT"Bütün adresleri taksim edi
niz";TAB(60);"(2)";PRINT
6050 PRINT"Seçilmiş adresleri ekrana
ekrana basınız";TAB(60);"(3):PRINT
```

```
6060 PRINT"Seçilmiş adresleri taksim
ediniz";TAB(60);"4":PRINT
6070 PRINT"Bir adresi ekrana basınız";
TAB(60);"(5):PRINT
6080 PRINT" bir adresi taksim ediniz";
TAB(60);"(6)":PRINT:PRINT:PRINT
6090 PRINT CHR$(7);:PRINT"İstenilen nu
marayı kaydediniz"
6100 aX=INKEY$:a=VAL(a$):IF a<1 OR>a 6
THEN 6100
6110 MODE 1:IF a/2=INT(a/2) THEN n=8
6120 ON a GOSUB 7000,7000,8000,8000,90
00,900
6130 n=OPRINT"Diğer beyanları istiyorm
usunuz? (e/h)"
6140 a$=INKEY$: IF a$="e" THEN 6000 EL
SE IF a$<>"h" THEN 6140
6150 RETURN
7000 REM
7010 REM *** Diğer adreslerin beyanı***
7020 FOR index=1 TO gösterge
7030 GOSUB 20000
7040 NEXT index
7050 RETURN
8000 REM
8010 REM *** Seçilen adreslerin beyanı
***
8020 PRINT CHR$(7);:PRINT" sabit bir s
oyad veya oturduğu yerin adresi basılı
yor mu?'N' veya'W' yi kaydediniz": PRI
NT:PRINT
8030 a$=INKEY$:a$=UPPER$(a$):IF a$="W"
THEN 8090 ELSE IF a$<>"N" THEN 8030
8050 For index=1 TO gösterge
8060 IF adı$=adı$(index) THEN GOSUB 20
000:a$="bulundu"
```

```
8070 NEXT index
8080 GOTO 8130
8090  PRINT  CHR$(7);:INPUT  "bulunması
gereken ikamet yerinin adı nedir";yer$
8100 FOR index=1 TO gösterge
8110  IF  yer$=yer$(index)  THEN  GOSUB
20000:a$="bulundu"
8120 NEXT index
8130 a$<>"bulundu"THEN PRINT:PRINT:PRI
NT" hiçbir adres mevcut değildir":PRIN
T
8140 RETURN
9000 REM
9010 REM *** Bir Adresin Basılması ***
9020 PRINT CHR$(7);:INPUT"Adresi veril
mesi gereken şahsın ismi nedir (ön isi
m,soyad)";ön isim$,isim$:PRINT:PRINT
9030 FOR index=1 TO gösterge
9040 IF ön isim$=ön isim$(index)AND is
im$=isim$(index) THEN GOSUB 20000:GOTO
9070
9050 NEXT index
9060 PRINT"Adres"ön isim$;" ";isim$:PR
INT "mevcut değil":PRINT
9070 RETURN
10000 REM
10010 REM ***Bir adresin kaydedilmesi*
**
10020 PLRINT:PRINT
10030 PRINT CHR$(7);:INPUT"Hitap";hita
p$(index)
10040 PRINT CHR$(7);:INPUTön isim";ön
isim$(index)
1050 PRINT CHR$(7);:INPUT "soyad";soya
$(index)
10060 PRINT CHR$(7);:INPUT"cadde
```

```
";cadde$(index)
10070 PRINT CHR$(7);:INPUT"Ev numarası
";numara$(index)
10080 PRINT CHR$(7);:INPUT "Posta Kutu
su";posta kutusu$(index)
10090 PRINT CHR$(7);:INPUT "Posta kodu
";posta kodu$(index)
10100 PRINT CHR$(7);:INPUT"ikamet yeri
";yer$(index)
10110 PRINT CHR$(7);:INPUT" Telefon ko
du";kod$(index)
10120 PRINT CHR$(7);:INPUT "Telefon nu
marası";telefon$(index)
10130 PRINT:PRINT
10140 RETURN
20000 REM
20010 REM *** Bir adresin verilmesi***
20020 PRINT# n
20030 IF hitap$(index)<>" " THEN PRINT
#n,hitap$(index)
20040 IF ön isim$(index)<>" " THEN PRI
NT#n,ön isim$(index);" ";
20050 IF isim$(index)<>" "THEN PRINT#n
n,nisim$(index)
20060 IF cadde$(index)<>" " THEN PRINT
#n,cadde$(index);" ";
20070 IF numara$(index)<>" " THEN PRIN
T#n,numara$(index)
20080 IF posta kutusu$(index)<>" " THE
N PRINT#n,"posta kutusu";posta kutusu$
(index)
20090 IF kod numarası$(index)<>" " THE
N PRINT#n,kod numarası$(index);" ";
20100 IF yer$(index)<>" "THEN PRINT#n,
yer$(index)
20110 IF kod$(index)<>" " THEN PRINT #
n,Tel(";kodX(index)?")";
20120 IF telefon$(index)<>" " THEN PRI
```

```
NT# n,telefon$(index)
20130 PRINT#n
20140 RETURN
```

100 Program ismi
110 Ön yükleme
120 Adres kapasitesinin belirtilmesi
130 Bilgi matrislerinin boyutlandırılması
170'den
190'a kadar program isminin verilmesi
200'den
280'e kadar ana liste metninin verilmesi

290 Arzu edilen fonksiyonun soruşturulması
300 Fonksiyonun çağrılması yanlış adreslerden dolayı maksada uymadığında, ana liste metninin yeniden verilişi
310 Arzu edilen alt programın belirlenmesi
320 Metin basımı
330 Program sonu

1000 Adreslerin kaydedilmesi için alt program
1020'den
1030'a kadar metin basımı
1040 Adres göstergesinin bir yükseltilmesi ve alt program değişkenleri için bilgi göstergesinin belirtilmesi
1050 Adreslerin kaydedilmesi
1060 Metin basımı
1070 Soruşturma durumu
1080 Ana listeye geri dönüş

2000 Adreslerin tanzim edilmesi için alt program
2020 Adresi değişmesi gereken şahsın isminin kaydedilmesi
2030 Adreslerin geçişi için döngü başlangıcı
2040 Değiştirilmesi gereken adres bulununca: döngüden çıkış
2050 Adreslerin geçişi için döngü sonu
2060 Arama döngüsünün başarısız sonuçlanması durumunda metin basımı ve bir sonraki program satırlarına atlama
2070 Değiştirilmesi gereken adresin verilmesi
2080 Değişen adresin kaydedilmesi
2090 Metin basımı
2100 Soruşturma durumu
2110 Ana listeye geri dönüş

3000 adreslerin silinmesi için alt sistem
3020 Adresi silinmesi gereken şahsın isminin kaydedilmesi

3030 Adreslerin geçişi için döngü başlangıcı
3040 Silinmesi gereken adres bulununca: döngüden çıkış
3050 Adreslerin geçişi için döngü sonu

3060 Arama döngüsünün başarısız sonuçlanması durumunda metin basımı ve daha sonraki döngüye atlama

3070 Silinmesi gereken adresin yerine adreslerin yerleştirilmesi için döngü başlangıcı
3080'den
3170'e kadar adreslerin yerleştirilmesi
3180 Silinmesi gereken adresin yerine adreslerin yerleştirilmesi için döngü sonu

3190 Göstergenin bir indirgenmesi ile adres göstergesinin düzeltilmesi
3200 Metin basımı
3210 Son adres silinince alt programın sona ermesi
3220 Metin basımı
3230 Soruşturma durumu
3240 Ana listeye geri dönüş
4000 Adreslerin okunması için alt program
4020 Metin basımı

4030 Okuma verisinin açılması
4040 Adres göstergesinin okunması

4050 Adreslerin okunması için döngü başlangıcı
4060'dan
4150'ye kadar adreslerin okunması
4160 adreslerin okunması için döngü sonu

4170 Okuma verisinin kapatılması
4180 Ana listeye geri dönüş

5000 adreslerin okunup bitirilmesi için alt program
5020 Metin basımı

5030 Okutma verisinin açılması
5040 Adres göstergesinin okunup bitirilmesi

5050 Adreslerin okunup bitirilmesi için alt program
5060'dan
5150'ye kadar adreslerin okunup bitirilmesi

5160 Adreslerin okunup bitirilmesi için döngü sonu

5170 Okutma verisinin kapatılması
5180 Ana listeye geri dönüş

6000 adreslerin verilmesi için alt program
6020'den
6090'a kadar liste metninin verilmesi

6100 arzu edilen fonksiyonun soruşturulması
6110 Verme biriminin belirtilmesi
6120 arzu edilen alt programın belirlenmesi
6130 Metin basımı
6140 Soruşturma durumu
6150 Ana listeye geri dönüş

7000 Bütün adreslerin basılması için alt program
7020 Adreslerin basılması için döngü başlangıcı
7030 Bir adresin basılması
7040 Adreslerin basılması için döngü sonu

7050 Listeye geri dönüş

8000 Seçilen adresin basılması için alt program
8020 Metin basımı
8030 Soruşturma durumu
8040 Soyadının kaydedilmesi

8050 adreslerin geçişi için döngü başlangıcı
3060 Aranan adresin bulunması durumunda: adresin basılması
8070 Adreslerin geçişi için döngü sonu

8080 Bir sonraki döngüye atlama

8090 Oturma yerinin kaydedilmesi

8100 Adreslerin geçişi için döngü başlangıcı
8110 Aranan adresin bulunması durumunda: adresin basılması
8120 Adreslerin geçişi için döngü sonu

8130 Arama döngüsünün başarısız sonuçlanması halinde metin basımı
8140 Listeye geri dönüş

9000 Bir adresin basılması için alt program
9020 Adresi basılması gereken şahsın isminin kaydedilmesi

9030 Adreslerin geçişi için döngü başlangıcı
9040 Aranan adresin bulunması halinde:
Adresin basılması ve döngüden atlama
9050 Adreslerin geçişi için döngü sonu

9060 Arama döngüsünün başarısız sonuçlanması halinde metin basımı
9070 Listeye geri dönüş

10000 Bir adresin kaydedilmesi için alt program
10020'den
10130'a kadar bir adresin kaydedilmesi
10140 Ana programa geri dönüş

20000 Bir adresin verilmesi için alt program
20020'den
20130'a kadar bir adresin mevcut bütün kısımlarının verilmesi
20140 Ana programa geri dönüş

3.9 Sıralama Programları (Yordamları)

Yüksek bir bilgi işleme hızına sahip olduğu için, büyük bilgi yığınlarının sıralanmasında bilgisayar oldukça uygundur.

Bu güne kadar sayıların bilgisayarla düzenlenmesi için yüzlerce metod bulunmuştur. Bu bölümde bunlardan üç eski ve kısa metod anlatılacaktır.

Bu üç program da "alan" sayı alanını "sayı" değerleri ile sayıların büyüklüğüne göre tasnif eder, öyle ki sondaki en küçük sayı başta, en büyük sayı ise alanın son pozisyonunda bulunur.

En basit tasnif algoritması, isminden de anlaşılacağı gibi, EASY-SORT'dur.

```
10 REM *** EASY SORT***
20 FOR lauf=1 TO sayı
30 minumum=1E+38:position=0
40 FOR index=lauf TO sayı
50 minumum>alan(index) THEN minumum=al
an(index):Position=index
```

```
60 NEXT index
70 değişim=alan(lauf):alan(lauf)=alan
(position):alan(position)=değişim
80 NEXT lauf
```

"Lauf" değişkenli dış döngü, "index" değişkenli iç döngünün her çevrimden sonra sayı "alan"ının daha önce çalışmasını sağlayan bir pozisyonla ilgilenir. Bu şekilde iç döngünün sıralama sayılarına dokunulmaz.

İç döngü başlamadan önce, bu döngünün değişkenlerine ön yükleme yapılır. "Minimum" değişkenine en yüksek değer verilir ve "pozisyon" değişkeni sıfıra getirilir.

Bununla iç döngü, aktüel "index" değişkeniyle ifade edilen alan pozisyonunun şimdiye kadar oluşan en küçük değeri içerip içermediğini kontrol eder. Durum böyle ise "minimum" değişkeninin değeri ve "pozisyon" değişkeninin durumu koordine edilir.

İç döngü sona erdiğinde, son çevrimdeki iç döngü tarafından yakalanan alanın ilk elemanı, en küçük değeri içeren alanın elemanıyla değiştirilir. Bu şekilde alan başlangıcına en küçük sayı rastlar ve alanın sonuna en büyük değer sıkıştırılır. Dış döngü tamamen çevrildiğinde, sayı "alan"ı sıralanır.

Daha sonraki sıralama metodu BUBBLE-SORT denilen metoddur.

```
10 REM *** BUBBLE SORT ***
20 FOR lauf=2 TO sayı
30 FOR index=sayı TO lauf STEP-1
40 IF alan(index-1)>alan(index) THEN d
eğişim=alan(index):alan(index)=alan(in
dex-1):alan(index-1)=değişim
50 NLEXT
50 NEXT index
60 NEXT lauf
```

İç ve dış döngü tekrar oluşur. Bu kez dış döngü, sayı "alan"ının arkasında çalışan iç döngünün her çevrimde, daha önceki çevrimi sona erdiren pozisyonla ilgilenir. Böylece devamlı çalışma alanının sıralı elemanları iç döngüyle kapalı bırakılır.

İç döngüde, alanın geçen durumunun değerinin "index" değişkeniyle anlatılan aktüel durumunun değerinden daha büyük olup olmadığı gözden geçirilir. Bu karşılaştırma olumlu (pozitif) ise, her iki eleman karşılıklı değiştirilir.

Böylece iç döngünün her çevrimiyle alan başlangıcına küçük bir sayı rastlar. Bu sayı kesin pozisyonunu alır, bu da BUBBLE-SORT'tur.

İç ve dış döngü sona erdiğinde sayı "alan"ı komple sıralanır. Son sıralama metodu RIPPLE-SORT'tur.

```
110 REM *** RIPPLE SORT ***
20 flaf=0
30 FOR index=1 TO sayı-1
40 IF alan(index)>alan(index+1) THEN
değişim=alan(index):alan(index+1):alan
(index+1)=değişim:flag=1
50 NEXT index
60 IF flag=1 THEN 20
```

Bu sıralama programının başlangıcında "flag" göstergesi sıfıra getirilir. Böylece ortak sayı "alan"ını çeviren döngü başlar.

"index" değişkeniyle saptanan alan pozisyonunun değeri, takip eden durumun değerinden daha büyük ise, her iki eleman birbiriyle değiştirilir ve "flag" göstergesi bire getirilir.

Döngü sona erdikten sonra, sayı değişiminin yani "flag" göstergesinin oluşup oluşmadığı kontrol edilir.

Değişim oluştuğunda döngü yinelenir. Komşularıyla değiştirilmesi gereken elemanlar yeniden aranır.

Sıralama döngüsünün çalışması uzun sürede tamamlandığında, komple döngü çevrimi tek bir değişim olmaksızın başarılı ve sayı "alan"ı tamamıyla sıralanır.

Bir sıralama programının hızı onun en önemli başarı kriteridir. Bu yüzden, rastgele dokuz rakamlı yüz değerden oluşan sayı "alan"ının sıralaması için üç programın karşılıklı durumu izlenir.

EASY-SORT: 27,5 saniye
BUBBLE-SORT: 60-65 saniye
RIPPLE-SORT: 90-110 saniye

Gördüğünüz gibi EASY-SORT yalnızca basit değil bilâkis aralıklı en hızlı metod'dur. Her şeyden önce bu sadece küçük, sıralamasız alanlar için geçerlidir.

EASY-SORT sürekli olarak toplam alanları birbiriyle karşılaştırır, bu yüzden lauf süreci de kısmen özenle açıklanır.

Diğer her iki sıralama metodlarında lauf sürecinin değerleri değişir. Bunun sebebi ise kullanılan sayı "alan"larının tesadüfen az veya çok ön sıralamaya tâbi tutulmasıdır.

Sayı "alan"ının ön sıralaması kabaca yapıldığında, BUBBLE-SORT ve özellikle RIPPLE-SORT metodu yararınıza olacaktır.

BUBBLE-SORT değiştirme operasyonunun büyük bir sayısını biriktirir, çünkü RIPPLE-SORT her sıralama çevrimi için, sayı "alan"ının düzenlenip düzenlenmediğini hafızaya alır, eğer hiç bir sıralama yoksa işlevi hemen sona erer.

Şayet alan tam olarak düzenlenirse, EASY-SORT da 27,5 saniyede sıralanır.

Bu üç sıralama metodunun yanında daha birçok metod vardır. En başarılı sıralama programı, farklı sıralama algoritmalarının bir birleşiminden oluşur ve kendi fonksiyonu içinde alışılmışın dışında zor olarak anlaşılır.

Bu konu daha sonraki literatur'da ayrıntılıca tartışılmıştır.

3.10 Metin İşleme

Konforlu metin işleme sistemleri mektupların ve manuskriptlerin düzenlemesi sırasında kullanan kişiye zaman kazandırır. Bu görev için şahsi bilgisayarlar tercih edilir.

CPC 464'de donanımsal özelliklerinden dolayı (profesyonel tuş takımı, her satırda seksen karakterlik ifade) metin izlemede kullanılmaya elverişlidir.

Burada, gerekli kullanım örnekleri ve dökümantasyonlarıyla komple metin işleme sistemini yeniden basmak (kopya etmek) anlamsızdır, çünkü bu kitabın çevresi (etki alanı) daralabilir. Bu dar çerçeve de CPC 464'le metin işleme programı olmaksızın metinlerin işlenmesi mümkündür.

Program satırlarını PRINT komutlarıyla kullanmak suretiyle bilgisayar üstünde bir metin düzenleyebilirsiniz. İlâve program satırlarının eklenmesiyle metni tamamlayabilir, satırların silinmesiyle kısaltılabilir. Ayrıca her program satırı edide edilebilir ve değiştirilebilir. Bu şekilde uzun metinleri bilgisayarda düzenlemek ve kasetlerde depolamak mümkündür. Peki metni basıcıya nasıl verirsiniz?

Birçok ev bilgisayarlarında PRINT komutlarını, basıcıya bir metni vermek için, LPRINT komutlarıyla değiştirmelisiniz. CPC 464'te ise bu gereksizdir.

Bölüm 3.1'deki "içindekiler" programı PRINT # n', komut zincirli, bir fonksiyon tuşu kapsadığı bilinmektedir. Bu komut zincirinin üstünlüğü daha sonraki bütün işaretlerin program satırlarının sonuna kadar hem ekran üzerinde hem de bağlanmış bulunan basıcı üzerine verilebildiğidir. Bu komut zincirinde "n" değişkeniyle temsil edilen giriş/verme birimi belirtilmiştir. Şayet "n"i tanımlamazsanız, değer otomatik olarak sıfır olacak ve komutu takip eden metin ekrana verilecektir. Buna karşın "n" değişkenine sekiz değeri verilirse, metin basıcı üzerine verilir.

Metinlerin düzenlenmesi metodu aşağıdaki gibidir:

AUTO komutuyla CPC 464'ten satır numaralarını ürettirebilirsiniz, CTRL ENTER komutu gerekli komut zincirini verir ve bu durumda sadece metni kaydetmeniz gerekmektedir.

Program satırının sonunda tırnak işaretini koyabilirsiniz.

Aşağıdaki liste bu şekilde düzenlenen ve n=8 ile GOTO 10'un basıcıya verilebileceğini gösteriyor.

```
AUTO 10
10 PRINT#n,gösteri mektni
20 PRINT#n
30 PRINT#n,"Bu küçük örnek canlandırıl
malı,tıpkı
40 PRINT # n,CPC 464 deki metinsiz
metin işleme sistemi gibi
50 PRINT#n,"uzun metinler yerleştirilm
eli ve kapalı basıcıya
60 PRINT#n,"verilmelidir.
70 PRINT#n,"Program satırlarının eklen
mesi ve silinmesiyle
80 PRINT#n,"metin değiştirilir ve düzeltilir,
tıpkı
90 PRINT#.n,"CPC 464'ün edite imkanlarının
kullanılması gibi
```

Bu örnekte her metin satırı bir program satırını gerektiriyor.

Her satıra 40 veya 80 karakterli metinler yazarsanız, daha da kolay olur, çünkü bu durumda ekranda 1 veya 2. Mod'da hangi ifadenin metin satırına uyduğunu, nerede ayrılması gerektiğini görebilirsiniz. 8 veya 4 metin satırını bir program satırına sığdırmanız mümkündür. Aşağıdaki örnek her satırı 40 karakterli bir metni gösteriyor.

```
AUTO 100
100 PRINT#n,"gösteri metni
110 PRINT#n,
120 PRINT# n,"bu örnek programla birçok metin
satırının bir program satırına sığdırılabilece
ğini görebilirsiniz.
```

```
130 PRINT#n,"Bu metod eğer siz 40 veya 80
karekterli her satırı kullanırsanız çalışır.Bu
şekilde 1.ve 2.moddaki metin satır genliği.ekr
an genişliğine uyar ve bir metnin ne zaman
140 PRINT#n,"sona erdiğini ve satırların
nerede ayrıldığını bilirsiniz.Bu metod çok
basit ama etkilidir.Bununla kitabın manuskript
i önceden belirlenen metodla oluşur.
```

3.11. Oyunların Programlanması

Zamanla ev bilgisayarları büyük ölçüde video oyunlarına taraftar olmuştur. Çünkü zamanla her oyun heyecanını yitirir, pazarlarda yeni oyunlara ihtiyaç sürekli olarak artar. Bazı programcılar oyun programlarının düzenlemekle iyi bir kazanç elde ederler.

Bir bilgisayar sahibi olarak oyun programlanmasının tekniği hakkında biraz bilgi edinirseniz daha yararlı olur. Belki de yılın en sevilen oyununu, yazmayı ve pazarlamayı başarırsınız! (Bunun yanında sizin milyonlar kazanacağınız garanti edilemez. Şansınız yoksa programınız illegal olarak çoğaltılır ve satılır).

Seri programları ayırmak için oyunlar da grafikle çalışılır. İyi bir grafik iyi bir oyun için garantidir. Bu yüzden oyun programlayıcısı olarak bilgisayarınızın grafiğini iyi yapmanız ve yerleştirmeniz çok önemlidir. Tüm olanakların kullanılmasında oyununuzun rekabeti ortaya çıkar. (Oyunun konusunun ilginç olduğu açıklanmalıdır).

Grafik bir oyun programında önemli bir yer değeri tuttuğundan, bu bölümde de özel bir yer almıştır.

Diğer programlarda olduğu gibi, bir oyun izenle plânlanmalı, anlamayı kolaylaştırmak ve özel program bölümünün değişimi basitleştirmek için yapı oluşturmalıdır.

Oyun programı düzenlenmeden önce birçok ana bloğa ayırmak anlamlı olur. Bunlar:

Başlık resmi
Oyun idaresi
Oyun alanı
Oyunun işlevi
Oyunun sonu

Oyunun bölümlerine, özel bir program için önemli bilgiler ve teşvikler bölümünü gösterebilirsiniz.

3.11.1 Çözüm, Renkler, Güçler

İyi bir grafik çözümüyle çalışabilmek için, en iyisi sıfır çok renkli modu unutunuz ve dört renkli bir numaralı standart modu seçiniz. Dört renk her oyun için yeterlidir, çünkü her sahne değişiminde yeni renkler kullanılabilmektedir.

Ayrıntılarla ilgilenenler 2 numaralı grafik modunu kullanmalıdır. Bu modda bir futbol programının oyuncularının triko reklâmı kolayca okunabilir.

Şaka bir yana, 2 numaralı mod iki olanaklı renklere uyar, ama 80 karakterli her satır matematiksel tasvir ve metin işlemesi için oyun programından daha önce gelmektedir. 2 renkli bir oyun uzun süre sıkıcı ve cansız bir etki bırakır. Oyuncu sürekli olarak yeni grafik efektleriyle sürpriz yapmalıdır.

Renkler

Eğer renkli bir monitör için düşünülen bir oyun renksiz bir monitörde görüldüğünde, zevkini heyecanını önemli ölçüde yitirir. Programın yazarının, renk bileşiminin tek renkli bir ekranda nasıl etki bırakacağını düşünmediği bir gerçektir. Renkler ve gerçekler arasındaki zıtlıkların sonucu, iki farklı rengin tek renkli bir monitörde aynı rengin tonlarını verebileceğidir.

Programlama anında renklere dikkat ediniz, CPC 464'ün renk değerleri dama şeklinde yapılmamalı, aksine en koyudan en açık renk tonuna kadar renksiz bir monitörde açıkça nüansı verilmelidir.

Bır oyunda benzer veya zıt renklerin mümkün olduğunca büyük renk değer farkıyla seçilmesi anlamlı olur. Pratikte beş minimal fark korunmalıdır (Bakınız. bölüm 2.4.2)

CPC 464'te mümkün renkleri iyi düzenlerseniz, renk farklarını ayırabileceğiniz altı büyük grup elde edersiniz. Bu gruplar sarı, kırmızı, eflâtun, mavi, turkuvaz ve yeşille ifade edilir. Oyun programı için bir gruptan en sevdiğiniz rengi arayınız ama bir renk grubunun birkaç renk tonunu kullanmayınız, aksi takdirde resim can sıkıcı hale gelir ve bu şartlar altında oyuncunun renk farkı dikkate alınmaz.

Güçler

Bir güç aşağı yukarı 20 x 20 boyutlu bir resmin grafik işaretidir. Bu küçük insanların ve uzay gemilerini anlatmak için yeterlidir. Güçün yararı tek bir komutla çözülmesi ve işletilmesidir. Çünkü hareketli grafiğin tasviri çok koordinatlı bir maddenin (nesnenin) çözülmesinin ve yerleştirilmesinin bir sonucundan oluşmaktadır, güç tekniği hareketli grafiğin programlanmasını son derece kolaylaştırır.

Malesef CPC 464'le güçleri anlatmak, çözümlü grafikli ev bilgisayarlarına nasıl alışılacağını anlatmak mümkün değildir.

SYMBOL komutlu bir grafik işaretinin düzenlenmesi kabul edilebilir bir alternatif sunar. Bu işaretler yanyana bulunan büyük nesneleri temsil etmektedir ve TAG und TAGOFF'a grafik koordinatları hakkındaki görevi, hareketlerin üretilmesini mümkün kılmaktadır.

Bu şekilde hareketli grafiği üretmek için grafik işaretlerini geriye almalı ve ASCII-CHRS(32) işaretli eski durumları (pozisyonları) silmelisiniz.

Akıcı bir hareket işlevi için ROM programlarını CALL 48409 komutuyla çağırırsanız, hareket gerçekleşmiş olur.

Aşağıdaki program örneği 4 standart grafik işaretinden oluşan kareyi, soldan sağa doğru yavaş yavaş ekrana kaydırır.

```
10 MODE 2:CLEAR:TAG:ORIGIN 0,200
20 a=212:b=213:c=214:d=215
30 CALL 48409:MOVE x,y:PRINT CHR$(b);CHR
$(a);
40 CALL 48409:MOVE x,y+16:PRINT CHR$(c);
CHR$(d);
50 FOR lauf=1 TO 50:NEXT lauf
60 MOVE x,y:PRINT"  ";
70 MOVE x,y+16:PRINT"  ";
80 x=x+4:IF x=644 THEN x=0
90 GOTO 30
```

Grafik işaretleri program işlevini hızlandırmak ve sembolün problemsiz bir değişimini mümkün kılmak için değişken olarak tarif edilir.

3.11.2 Başlık Resmi

Programın yüklenmesinden sonra ekranda görünen başlık resmi bir oyun programının ilk tesirine aracılık eder. Bu başlık resmi bir kitabın cildine veya bir plâğın kutususuna uymaktadır. Bundan dolayı özel özen gösterilmelidir, çünkü oyun oyuncuya hitap etmeli ve oyuna teşvik etmelidir.

İyi bir başlık resmini teşvik eden çok şey vardır. İlk olarak program adı, programlayıcının adı ve kayıtlar. Ayrıca oyuncunun yeni değişime hakim olup olmadığını hemen görebilmesi için tamamlamanın tarihi veya programın numarası bulunmalıdır.

SYMBOL komutu ile kendi işaret cümlelerinizi tarif edilebilir ve metni harflerle basabilirsiniz. Program isminin üç boyutlu yazılarla belirtilmesi daha iyidir.

Fakat iş bu kadarla kalmaz. Üzerinde başlığı, yazarı ve yayınevi basılı olan bir kitabın kapağını tasavvur ediniz. Kitabın kapağı her ne kadar sıkıcı olsa da kitabın içeriği o kadar enteresan olmalıdır.

Görüyorsunuz ki grafik bir programın başlık resmini oluşturur, nihayet oyununuzun elementleri sadece harflerden meydana gelmez.

Tesirli olan, hareket eden bir grafiktir, örneğin değişen bir resim çerçevesi veya oyununun birkaç elementi (istiap hacmi vs.) Bunlar ise size oyun programınızın grafik ifade kalitesini göstermek için.

Oyununuzun başlık resmi metin ve grafikten oluşur. Kullanıcı ilk optik etkiyi bu başlık resmi ile elde eder. Daha neler bulunmalıdır? Evet, müzik! Neşeli bir melodi ile başlık resminin fon müziği sizi yumuşatır, oyununuzun sadece optik olmadığını aksine akustik de olduğunu size gösterir. ayrıca melodiniz ses darbelerinden oluşur ve böylece tanınmış oyunlar insanın kulağında kalır ve oyunun reklâmını yapar.

3.11.3 Oyun Kılavuzu

Oyun kılavuzu programlayıcının çoğunlukla dikkat etmesini isteyen bir oyunun bölümüdür.

Bir oyun kılavuzu açık ve belirli yazılmalı ve bir oyunun değişimi açıklanmalıdır, aksi halde faydalı bilgiler bulunmadığı için kullanıcının hakim olamadığı bir durumda bulunulur.

Siz noksan ve oyun kılavuzunun bulunmadığı oyun otomatiğini tanıyorsunuz. Oyuncu bilgilerini ya sorarak yada oynayarak öğrenmelidir. Bu boşuna didinmek ve programa kötü popularite demektir.

Kural olarak macera oyunları haricindeki bütün oyunlar nokta oyunları olduğu için, oyunun yöntemi hangi hareket için hangi durumda kaç noktanın gerekli olduğunu belirtmektedir.

Oyunun anlamı, oyuncunun hangi amaç için çalıştığını bilmesi açısından, etraflıca açıklanmalıdır. Noktaların açık bir şekilde toplanması ön plânda olmamalıdır. Müşkülâtlı oyunlarda oyuncu diğer oyunlara erişebilmek için, zamanla zor görevleri çözmeyi başaracaktır.

Ayrıca oyun alanında kullanıcıya tarif edilen player-missile oyununun nasıl kumanda edilebildiği (meselâ tuş takımı harfleriyle mi, yoksa joystick'le mi?) açıklanmalıdır.

Şayet oyun yönetiminiz bütün bunlara uyuyorsa, tam bir şekilde uygulanabilir.

3.11.4 Oyun Alanı

Oyun uygulanırken oyun alanı yada oyun alanları teşkil edilir. Şayet oyununuzda tek bir oyun alanı varsa ve arka plân bunu müteakip stasyoner ise, oyuncunun çabuk canının sıkılmaması için değer enteresan bir alana yerleştirilir. Belki de oyun alanının veri plânını tesadüfen terkedebilirsiniz, bu durumda oyununuzun konusu mümkün olduğu kadar diğer bir şekilde görünecektir.

Bunun aksine oyununuzun oyun alanı hareketli ise yada sık sık değiştirilirse, bu nokta için fazlaca bir zahmete gerek yoktur.

Genellikle bir oyun alanı iki kısımdan oluşur: Manzara ve gösterge birimi. İlk kısım kullanıcıya şu anda nerede olduğunu gösterir, ikinci kısım oyuncunun şu andaki durumunu yada hareketinin şu andaki durumunu (meselâ hızını) belirtir. Gösterge birimi verileri mümkün olduğunca grafik şeklinde belirtir.

Kalan cephane yada akaryakıt gibi gerçekten önemli bilgiler gösterilebilir, aksi halde oyuncu birçok göstergenin içersinde şaşırır kalır.

3.11.5 Oyunun İşlevi

Bir oyunun en önemli bölümü oyunun işlevidir. Bu nokta değişken olduğu ve oyunun konusu sizin hayalinize kaldığı için bu hususun belirtilmesi gerekir.

Şimdi burada esas bilgileri izleyelim:

Oyun olayı için karekteristik olan bir melodiyi paralel olarak olaya sokması video oyunlarında elde edilebilir. Bir oyunun objeleri çatışırsa, genelde bu ele-

mentlerden birinin en azından tahrip edilmesine veya yok olmasına yol açar. Bu aksiyona özel bir ses, gürültü ve diğer melodiler de karışabilir.

Sesin ve tonun paralel ifadesi CPC 464'le kolayca elde edilebilir. Kanaldan tonu bekleme dizisi tamamen boş ise, ONSQ (kanal) GOSUB komutu alt programa atlar. Grafik operasyon esnasında melodinin kesilmemesi için, tonlar tam zamanında verilmelidir. Sonradan yükleme öyle hızlı olur ki, oyuncu hiç bir şey farketmez.

Şu anda mevcut olan melodi CLEAR komutuyla kesilebilir. Bütün diğer veriler de beraber silindiğinden dolayı bu şüphesiz ki bir program için uygun olmayan bir metod'dur. Diğer bir alternatif ise uygun kanal üzerinde ses şiddetinin tonuna tam dolu olan bir Bit ile sıfırlamaktır. Tam dolu olan bir Bit ton verilmesini keser ve halihazırdaki tonu verir. Bu ton belirtilen ses şiddeti ile işitilemediğinden, bu metod prensipte ses kesilmesi işlemidir.

Genelde bir melodinin sona ermesinden sonra başka bir gürültü işitilmektedir, öyle ki bunu takip eden SOUND komutu aynı kanal üzerindeki tonu, daha önceki SOUND komutunda olduğu gibi eski melodiyi silmek için tam dolu bir Bit ile belirler.

3.11.6 Oyunun Sonu

Oyun ilginç bir sonla bitirilmelidir, çünkü bir oyunun sonu oynayanın ilgisini çekmelidir. Her tesirli sonuç müsaittir, yalnız kuvvetli bir GAME OVER'in göz kamaştırması sizi vazgeçirir, çünkü bundan oynayan sık sık sıkılabilir.

Kısa bir yorumda oyunun başarısının bir değeri sizin oyununuzun sonunda eksik olmamalıdır. Bu oyuncuyu ve uygun kelime seçiminde -özellikle kötü bir söz kesme durumunda- oyuna katılanları da sevindirir.

Son nokta durumun haberi 10 oyuncudan 20 oyuncuya kadar aynı şekilde mecburi liste gibidir. Böylece son oyunda uygun başarıyı elde edebildiğine göre ismini bu

listeye aktarabilmek imkânı doğar (Her oyun komputere veya oyuna katılanlara karşı bir rekabet mücadelesidir ve her oyuncu iyi bir sonuçta emin bir tanımayı da tecrübe etmeyi ister).

Özel ve mutlak en yüksek nokta sayısı programda sağlam olarak yerleştirilmeli ve bilinmelidir, çünkü bir oyunun programlayıcısının en iyi nokta durumundan daha fazla üstün gelmesi önemli değildir. Yani burada parola geçerlidir. Programlayıcı olarak yeteri kadar iyi oyunlar yazılmazsa, en azından oyun iyi oynanmak istenir.

Oyunun sonunda yeni bir deneme yapmak için kullanıcıda büyük bir ilgi uyanır.

SONSÖZ

Bu bölümle oyun programlanması ile birlikte kitabın sonuna da geldik. Bu sebepten dolayı sonuç aşağıdaki gibidir:

Her gayretli oyuncu bir oyun programında, özellikle programa uygun kayıtlar seçtiğinde, nasıl eğlendirici ve neşeli yorumların olduğunu bilir. Oyun olayı için bu yorumlar bir oyunda çorbadaki tuz gibidir! Her şeyden önce oyunun akışı esnasında özel yüksek ve düşük başarıların değerinde ve oyunun sonucundaki sona ermiş değerde uygun kararlar makbuldur ve ideal olarak yerleştirilir. Bundan dolayı oyununuza sık sık eğlendirici fikirler katınız. Birkaç örnek:

"Evet, kör bir tavuk bir buğday tanesi bulur"
"Kapalı gözlerle oynarsanız daha iyi olur"
"sirkte palyaço olarak çalıştınız mı?"
"En kısa zamanda benden daha iyi olacaksınız"

Biraz hayâl biraz da hünerle uygun yerlerde grafik efektler ve güldürüler yapabilirsiniz. Meselâ, bir hamburger için reklâm yapan bir korkuluk yada şanssız oyuncunun canını alan bir şeytan. Esas olan sizin daima başarılı olmanızdır. Kullanıcının sıkılmaması için oyunlarınıza bir parça eylence katmalısınız. Tabii ki bu program stili daha fazla bir zaman ve 08/15'lik programlama yerinden daha fazla bir bellek alanı gerektirir. Oyununuzun sık sık tekrar edilmesi ve sevilmesi buna bağlıdır.

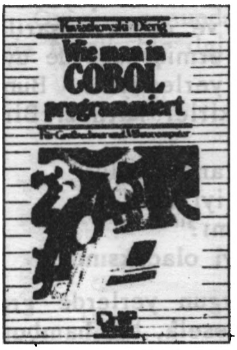
Şayet oyununuzdan ayrılamazsanız, gece gündüz bilgisayarla vakit geçirirseniz, bir Bestseller yazmışsınız demektir.

CPC 464'un donanım sisteminin kreatif etkisi için **yazar** tarafından plânlanan bir izleme bandı doldurulmuştur:

Bandın ismi "Hardware-Erwerterungen für den CPC 464 selbstgebaut"tur.

YAKINDA ÇIKACAK OLAN YAYINLARIMIZ





ATARI NELER YAPABİLİR?

Aptullah Kuzu'nun çevirisi ve VOGEL VERLAG yayınevi ile işbirliği sonucu bu konudaki ikinci yayınımızı yakında kitapçılarda bulabilirsiniz.

COBOL PROGRAMLAMA NASIL YAPILIR?

Dr.Nesrin Ertel'in çevirisi ve VOGEL VERLAG yayınevinin işbirliği ile yakında çıkacak olan bu kitap COBOL programlamayı öğrenmek ya da ilerletmek isteyenler için ideal bir yapıt.

BİLGİSAYARA GİRİŞ VE BASIC

Anadolu Üniversitesi Öğretim Üyesi Yrd Doç.Dr.Ali Güneş'in hazırladığı bu yapıt Temel Kavramlar, Bilgisayar yazılımı ve Donanımını ve BASIC programlama dilini içermektedir.

Bilimsel Araştırmalarda BİLGİSAYARLA İSTATİSTİK DEĞERLENDİRME VE VERİ ANALİZİ.

Anadolu Üniversitesi Öğretim Üyesi Doç Dr.Kazım Özdamar'ın hazırladığı bu yapıt özellikle İstatistikle uzaktan ya yakından ilişkisi olanlar için bir başvuru kitabı niteliğinde.

ATARİ-BASIC ÇIKTI.................ALDINIZMI?

I.Hamur baskı. 184 sahife olan bu kitap Atari bilgisayarı olanlar için Türkiye'de basılmış ilk Türkçe eser.

CPC 464 bilgisayarı, en çok satılan ev bilgisayarı olma yolundadır. Ancak, tecrübeli kullanıcıya da yardımcı olabilecek, anlaşılır ve kapsamlı bir kullanım kitabının eksikliği hissedilmektedir. Bu kitap, bilgisayarınızı daha iyi anlamanıza ve ondan daha çok yararlanmanıza yardımcı olacaktır.

Her CPC 464 bilgisayarı sahibinin pek çok yeni ve önemli bilgiler bulacağı bu kitap, CPC 464 kullanım broşüründen çok ileri kapsamlıdır. Kullanım broşürünü tamamen incelemiş olanların kafasında dahi birçok soru cevapsız kalmış olacaktır. Bilgisayarları için ek bilgi, yeni fikir ve program arayışı içinde olanların bu arzularını karşılamak üzere, bu kitap üç büyük bölüme ayrılmıştır.

Kitap, hareketli grafik ve sesli oyunların programlanmasına dair ilginç bir kesimle son bulmaktadır. Verilen programlar tümüyle denenmiş olup, bilgisayarın orijinal çıkışlarıdır. Bu nedenle hepsi CPC 464le kusursuz çalışacaklardır.

VOGEL-BUCHVERLAG
WÜRZBURG

Martin Aschoff **AMSTRAD CPC464 NELER YAPABİLİR**